An Antidissipative Transport Scheme on Unstructured Meshes for Multicomponent Flows

Bruno Després, Frédéric Lagoutière, Emmanuel Labourasse, Isabelle Marmajou

The International Journal on Finite Volumes
IJFV , **7**, *pp 30-65, (2010)*

An Antidissipative Transport Scheme on Unstructured Meshes for Multicomponent Flows

Bruno Després[†]
[†] *Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie, 75252 Paris Cedex 05, France*
despres@ann.jussieu.fr

Frédéric Lagoutière[⋆]
[⋆] *Département de Mathématiques, Bâtiment 425, Faculté des Sciences d'Orsay Université Paris-Sud 11, F-91405 Orsay CEDEX, France,*
and Project Team SIMPAF, INRIA Research Centre Lille-Nord Europe Park Plazza, 40, avenue Halley, F-Villeneuve d'Ascq CEDEX, France. Work supported by CEA/DIF.
frederic.lagoutiere@math.u-psud.fr

Emmanuel Labourasse[††]
[†] *CEA, DAM, DIF, Bruyères-le-Chatel, F-91297 Arpajon Cedex*
emmanuel.labourasse@cea.fr

Isabelle Marmajou[⋆⋆]
[†] *CEA, DAM, DIF, Bruyères-le-Chatel, F-91297 Arpajon Cedex*
isabelle.marmajou@cea.fr

Abstract

A new numerical method called Vofire is described here. It can be used for integrating transport equations approximated on general meshes in any dimension. The algorithm relies on a piecewise constant reconstruction of an unknown in each cell, where the reconstructed function varies only in the directions orthogonal to the velocity. This allows to design a non-dissipative and maximum preserving scheme for the transport of characteristic functions. It is then incorporated in a Lagrangian scheme with remeshing for the computation of multi-component compressible flows.

Key words : non-dissipative transport scheme, multidimensional reconstruction scheme, compressible multicomponent fluid flow, Lagrange + remap scheme.

1 Introduction

Advection of discontinuous profiles by means of Finite Volume schemes is still challenging despite constant works. Discontinuous profiles are common to an Eulerian CFD code dealing with multi-phase or multimaterial flows. This is also the case when ALE (Arbitrary Lagrangian Eulerian) simulations are performed. First introduced by Hirt *et al.* [13], ALE calculations consist in allowing the mesh to move with arbitrary velocity u independent of the flow velocity u_f. This method degenerates to the Euler framework when $u = 0$, and to the Lagrangian framework when $u = u_f$. The reader is invited to refer to the paper of Benson [2] for a comprehensive presentation, to [10] for an insight on ALE and to [12] for mesh regularization techniques.

In the case of Eulerian calculations using a Cartesian mesh, several interface reconstruction or tracking methods have been proposed (a review of the possibilities is available in [27]). These methods can be roughly grouped into three families: Front or Interface Tracking (see for instance [34, 25]), Level-Set [28], and Simple Line Interface Calculation (SLIC, see [24]) or Volume of Fluid (VOF, see [36, 23]). We know nothing in the literature about the use of Level-Set for 3D conservative calculations of multi-material flows in a Lagrange code, which is our ultimate goal in this work. This is why we do not discuss this method any more.

VOF seems to be the best candidate to calculate this kind of flows (see for instance [3] or [26] for a review on VOF methods). Unfortunately, VOF is quite difficult to extend to 3D unstructured meshes, in particular for more than two materials. Some promising attempts have been proposed (see for instance [11, 29, 31, 16]) but very few realistic simulations using these methods have been published.

We propose here a different way of avoiding the numerical diffusion of the interfaces. This method is based on the previous work [7], further extended to non-linear equations in [4] and in [19, 20] by one of the authors. See also [37] for the use of this approach in the context of Weno schemes. In [7] a flux-based anti-dissipative scheme is defined for high-speed compressible flow calculations, but which is by construction restricted to Cartesian meshes. This paper describes an extension of this scheme to 3D unstructured meshes. We concentrate on the design principle of the method and on the construction of an effective scheme. Theoretical considerations, such as obtaining fine a priori estimates and/or estimates of convergence, are not considered in this work. Instead we will rely on numerical tests to illustrate the gain of accuracy provided by the method proposed in the present work.

A basic model, convenient for expository purposes, is the transport equation of

a concentration $c \in [0,1]$ with velocity $\mathbf{u}$

$$\begin{cases} \partial_t c + (\mathbf{u}, \nabla c) = 0, & t > 0, x \in \mathbb{R}^2, \\ c(0,x) = c^0(x), & x \in \mathbb{R}^2. \end{cases} \tag{1}$$

The velocity $\mathbf{u} = \mathbf{u}(x)$ is a given smooth velocity field and the initial data is $c^0(x) \in [0,1]$. The velocity field can also be a function of time, but, for the sake of simplicity, we drop this dependency. The equation is equivalent to

$$\partial_t c + \operatorname{div}(c\mathbf{u}) = c \operatorname{div}(\mathbf{u}). \tag{2}$$

We will conduct the analysis first assuming that $\operatorname{div}(\mathbf{u}) = 0$, but then we will relax this assumption. An important issue for the numerical treatment of this PDE problem is the numerical diffusion. This phenomenon, easily understandable in dimension one, is more complex in higher dimension. It can be decomposed in two different types of diffusion. The diffusion of the first type, which is called longitudinal diffusion, is the one that occurs in the direction of the velocity. It is the diffusion present in one-dimensional computations. The second type, transverse diffusion, is typically multidimensional and is due to the fact that the fluid velocity is not aligned with the mesh. This distinction between the two phenomena could appear arbitrary, but is in accordance with basic numerical tests. For example consider an initial condition which is the characteristic function of a square $]0.25, 0.75[\times]0.25, 0.75[$. This profile is advected with the upwind scheme. The velocity direction $\mathbf{u}$ has a great influence on the result. It is illustrated on figure 1.

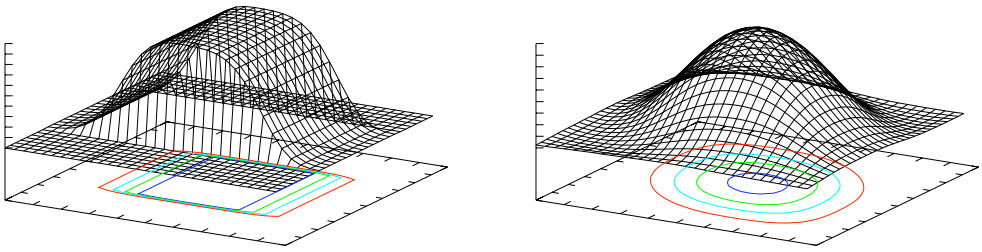


Figure 1: Upwind scheme. The initial condition is the characteristic function of a square. Final time $t = 1$. Periodic boundary conditions. On the left: the velocity $\mathbf{u} = (1,0)$ is aligned with the mesh; the result displays only longitudinal diffusion. On the right: the velocity $\mathbf{u} = (1,1)$ is not aligned with the mesh. The consequence is longitudinal and transverse diffusion.

The difference between longitudinal and transverse diffusions appears clearly in [6] which was a previous attempt to elaborate non-dissipative schemes on non-Cartesian grids. In what follows we construct new transport algorithms, which are based on the distinction between transverse diffusion and longitudinal diffusion.

The original feature of the schemes that we propose is the following. On a Cartesian grid directional splitting methods use the natural definition of the grid. But on a general grid this is not possible anymore. So in order to reduce the scheme

to a series of local one-dimensional problems (that one could solve efficiently) one needs to figure out what could be the local main direction, using some additional information. Usually one considers some gradient line of the solution, see [1, 2, 17]. In this work we propose to use the velocity direction. The method is based on the construction of anti-dissipative flux. So it is different from the MUSCL approach for which we refer the reader to the very complete review [1] (see also references therein). MUSCL reconstructions are piecewise-linear, while ours are piecewise-constant. Since the core of the method is the idea of reconstruction, we propose to call it **Vofire**, which stands for **Vo***lume* **fi***ni avec* **re***construction* in French (that stands for **Fi***nite* **Vo***lume with* **Re***construction*). Like MUSCL schemes, Vofire satisfies a local maximum principle in the form

$$\min\left(c_j^n, \min_{k\in N^-(j)} c_k^n\right) \le c_j^{n+1} \le \max\left(c_j^n, \max_{k\in N^-(j)} c_k^n\right)$$

where $N^-(j)$ is the set of all cells such that the flow is incoming from cell $k \in N^-(j)$ into cell j. This inequality is slightly more restrictive than the one studied in [1] where the minimum (resp. maximum) is taken over all neighboring cells.

We also show how to extend this method for multimaterial simulations. We prove the theoretical properties (maximum principle and conservativity) are essentially preserved in the remap step of a Lagrange+remap scheme. Indeed, the remap step of such a scheme can be recast in a numerical resolution of (1). In general the equivalent transport velocity does not satisfy $\mathrm{div}\,(\mathbf{u}) = 0$. But the transport of concentration c, $\partial_t(\rho c) + \mathrm{div}\,(\rho c\mathbf{u}) = 0$, where ρ is the fluid density that satisfies $\partial_t\rho + \mathrm{div}\,(\rho\mathbf{u}) = 0$, is equivalent to $\partial_t c + (u, \nabla c) = 0$. It allows to extend to the remap stage the analysis already performed for pure transport. Since the remap and the Lagrange steps are independent, there is no need to discuss the latter in this paper.

The plan is as follows. In section 2 we give the notations. The principles of the geometric reconstruction are developed in section 3. The Vofire Algorithm is explained in section 4. This is followed by pure transport numerical results in section 5. Then we show in section 6 a simple strategy to adapt this method in the remap phase of a multimaterial algorithm. The result of a challenging 3D numerical test is given in section 7. Finally we draw some conclusions about the advantages and limitations of VOFIRE.

2 Notations in 2D

We shall develop the schemes in the context of Finite Volume methods. We begin with notations, see figure 2. We consider a 2D mesh composed of open cells (T_j) such that $T_j \cap T_k = \emptyset$ for every j, every $k \neq j$. For all j, k, $l_{j,k}$ denotes the one-dimensional Lebesgue measure of the edge between adjacent cells j and k:

$$l_{j,k} = \mathrm{length}\left(\overline{T_j} \cap \overline{T_k}\right),$$

thus $l_{j,k} = l_{k,j}$. For each cell T_j, $N(j)$ denotes the set of indices of neighboring cells of T_j (cells having a common edge with T_j). For for $k \in N(j)$ (T_j and T_k

have an edge in common), $\mathbf{n}_{j,k}$ denotes the unit normal vector to the common edge directed outward from T_j. We thus have $\mathbf{n}_{j,k} = -\mathbf{n}_{k,j}$ for every $j \in \mathbb{Z}$ and every $k \in N(j)$. Let $\mathbf{u}_{j,k}$ be some consistent mean value of the velocity on the edge $\overline{T_j} \cap \overline{T_k}$. Then $N^+(j) \subset N(j)$ is the set of indices of the downwind neighbors of T_j and $N^-(j) \subset N(j)$ is the set of indices of the upwind neighbors of T_j

$$N^+(j) = \{k;\ (\mathbf{u}_{j,k}, \mathbf{n}_{j,k}) > 0\} \text{ and } N^-(j) = \{k;\ (\mathbf{u}_{j,k}, \mathbf{n}_{j,k}) < 0\},$$

where $(\cdot,\cdot)$ denotes the Euclidean dot product. Let $s_j = \text{area}\,(T_j)$ be the 2D Lebesgue measure of the cell T_j.

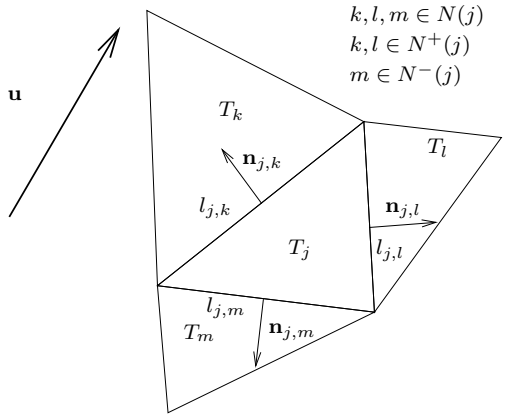


Figure 2: Mesh and notations (with $\mathbf{u}$ constant for simplicity).

Let $\Delta t > 0$ be the time step. We integrate the transport equation (2) over the time-space domain $[n\Delta t, (n+1)\Delta t] \times T_j$ and get

$$s_j \frac{c_j^{n+1} - c_j^n}{\Delta t} + \sum_{k \in N(j)} l_{j,k}(\mathbf{u}_{j,k}, \mathbf{n}_{j,k}) c_{j,k}^n = \left(\sum_{k \in N(j)} l_{j,k}(\mathbf{u}_{j,k}, \mathbf{n}_{j,k}) \right) c_j^n. \tag{3}$$

In this formula, the quantities $\mathbf{u}_{j,k}$ are approximate values of the given velocity $\mathbf{u}(x)$ on the edges

$$\mathbf{u}_{j,k} = \frac{1}{l_{j,k}} \int_{\overline{T_j} \cap \overline{T_k}} \mathbf{u} d\sigma, \tag{4}$$

and $c_{j,k}^n$ are approximations of the edge-based fluxes between times $n\Delta t$ and $(n+1)\Delta t$. This equation rewrites

$$s_j \frac{c_j^{n+1} - c_j^n}{\Delta t} + \sum_{k \in N(j)} l_{j,k}(\mathbf{u}_{j,k}, \mathbf{n}_{j,k})(c_{j,k}^n - c_j^n) = 0. \tag{5}$$

This formula defines a class of finite volumes schemes. For instance, the upwind scheme is obtained by taking $c_{j,k}^n = c_j^n$ for all $k \in N^+(j)$ and $c_{j,k}^n = c_k^n$ for all $k \in N^-(j)$. In this case (5) simplifies to

$$c_j^{n+1} - c_j^n + \frac{\Delta t}{s_j} \sum_{k \in N^-(j)} l_{j,k}(\mathbf{u}_{j,k}, \mathbf{n}_{j,k})(c_k^n - c_j^n) = 0, \tag{6}$$

where the sum is taken only over the edges corresponding to upwind neighbors. Note that this scheme satisfies the maximum principle under the Courant-Friedrichs-Lewy condition (CFL)

$$\nu_j = -\frac{\Delta t}{s_j} \sum_{k \in N^-(j)} l_{j,k}(\mathbf{u}_{j,k}, \mathbf{n}_{j,k}) \in [0,1]. \tag{7}$$

This upwind scheme is very diffusive and therefore useless for many computations.

3 Geometric reconstruction in 2D

We propose to use a geometric reconstruction to introduce some anti-dissipative mechanism in the scheme. We consider a triangular mesh in 2D, as in figure 3. In order to separate the problems of longitudinal and transverse diffusions we perform the reconstruction in two steps. For the sake of simplicity, we assume in this section that $\mathbf{u}$ is constant (this assumption will be removed in section 4). One has

$$\sum_{k \in N(j)} l_{j,k}(\mathbf{u}, \mathbf{n}_{j,k}) = \int_{\partial T_j} (\mathbf{u}, \mathbf{n})\, d\sigma = \int_{T_j} \mathrm{div}(\mathbf{u})\, dx = 0. \tag{8}$$

3.1 First step: the transverse reconstruction

Assume for expository purposes that the mesh is made with triangles in 2D. The transverse reconstruction consists in breaking a cell in two parts by a segment parallel to the velocity and modifying the value of the unknown in each of these two sub-cells. Each triangle T_j has at least one downwind neighbor and at most two. If it has only one downwind neighbor, we do not perform the transverse reconstruction (we do not cut the cell). Let us now assume that T_j has two downwind neighbors, T_k and T_l. It has then one upwind neighbor, T_m. We consider the intersection point of the two edges relative to the downwind neighbors and cut T_j along the line passing on this intersection point and parallel to $\mathbf{u}$. The two sub-cells are denoted $T_{j,k}$ and $T_{j,l}$: $T_{j,k}$ has T_k as (unique) downwind neighbor, and $T_{j,l}$ has T_l as (unique) downwind neighbor. This partitioning is illustrated on figure 3. We use symbols $s_{j,k}$ and $s_{j,l}$ to denote the areas of sub-cells $T_{j,k}$ and $T_{j,l}$ respectively. Clearly $s_{j,k} + s_{j,l} = s_j$ and $s_{j,k} > 0$ and $s_{j,l} > 0$. The aim is to define a reconstructed value $c^R_{j,k}$ in $T_{j,k}$ and a reconstructed value $c^R_{j,l}$ in $T_{j,l}$. We impose

$$s_{j,k} c^R_{j,k} + s_{j,l} c^R_{j,l} = s_j c^n_j \tag{9}$$

to guarantee the local conservativity. Let us write

$$\begin{cases} c^R_{j,k} = c^n_j + \lambda_{j,k} \left(c^n_k - c^n_j \right), & 0 \le \lambda_{j,k} \le 1, \\ c^R_{j,l} = c^n_j + \lambda_{j,l} \left(c^n_l - c^n_j \right), & 0 \le \lambda_{j,l} \le 1, \end{cases} \tag{10}$$

which means that $c^R_{j,k}$ and $c^R_{j,l}$ must satisfy a condition of local consistency. We introduce the idea of anti-dissipative schemes which will serve to find a unique value of $\lambda_{j,k}$ and $\lambda_{j,l}$.

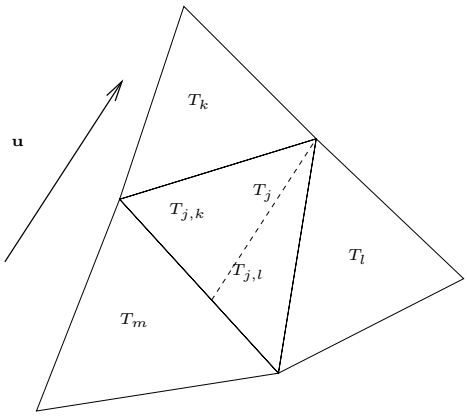


Figure 3: Transverse reconstruction.

REMARK 1 **A design principle for anti-dissipative methods** Our objective is to choose $\lambda_{j,k}$ and $\lambda_{j,l}$ in order to obtain an anti-dissipative scheme, with a very low level of numerical diffusion. This is the reason why we seek the largest possible $\lambda_{j,k}$ and $\lambda_{j,l}$ in the interval $[0,1]$. This design principle is inspired by many others numerical methods with low dissipation that have been published in the literature, see [38, 37, 7, 8, 4]. We will use this principle systematically in the rest of this work. But of course we cannot take $\lambda_{j,k} = \lambda_{j,l} = 1$ directly because the scheme has also to satisfy the maximum principle.

So our goal is to have the largest $\lambda_{j,k}$ and $\lambda_{j,l}$ in the interval $[0,1]$, but still satisfying the maximum principle. Equation (9) can be rewritten as $s_{j,k}c^R_{j,k}+s_{j,l}c^R_{j,l}-s_jc^n_j = s_{j,k}\left(c^R_{j,k}-c^n_j\right)+s_{j,l}\left(c^R_{j,l}-c^n_j\right)=0$, that is

$$\left[s_{j,k}\left(c^n_k-c^n_j\right)\right]\lambda_{j,k}+\left[s_{j,l}\left(c^n_l-c^n_j\right)\right]\lambda_{j,l}=0.$$

The solution of this equation is computed as follows.

1) If $s_{j,k}\left(c^n_k-c^n_j\right)s_{j,l}\left(c^n_l-c^n_j\right)\geq 0$**,** c^n_j is a local extremum in the transverse direction. Then we do not reconstruct, which means $\lambda_{j,k}=\lambda_{j,l}=0$ and

$$c^R_{j,l}=c^R_{j,k}=c^n_j. \tag{11}$$

2) If $-\dfrac{s_{j,k}\left(c^n_k-c^n_j\right)}{s_{j,l}\left(c^n_l-c^n_j\right)}>1$**,** the solution is obtained by taking $\lambda_{j,l}=1$,

$$c^R_{j,l}=c^n_l, \qquad c^R_{j,k}=c^n_j-\frac{s_{j,l}}{s_{j,k}}\left(c^n_l-c^n_j\right)=\left(s_jc^n_j-s_{j,l}c^n_l\right)/s_{j,k}. \tag{12}$$

3) If $-\dfrac{s_{j,k}\left(c_k^n - c_j^n\right)}{s_{j,l}\left(c_l^n - c_j^n\right)} < 1$**,** the solution is obtained by taking $\lambda_{j,k} = 1$,

$$c_{j,k}^R = c_k^n, \qquad c_{j,l}^R = c_j^n - \frac{s_{j,k}}{s_{j,l}}\left(c_k^n - c_j^n\right) = (s_j c_j^n - s_{j,k} c_k^n)/s_{j,l}. \tag{13}$$

PROPOSITION 1 Consider the finite volume scheme (5) with the fluxes $c_{j,k}^n = c_{j,k}^R$ defined by (11-13). The CFL condition on the time step to respect the maximum principle is (7), as for the upwind fluxes (6).

Proof The reconstructed quantities (10) respect the maximum principle. By construction the scheme is equal to a two steps algorithm: first step, use the upwind scheme for a mesh which is locally cut in smaller cells, as it is described in figure 3, and with cell quantities equal to the reconstructed quantities; second step, project onto the original coarse mesh. Therefore it is sufficient to check that the CFL condition (7) is the same for the original mesh (4 cells in figure 3) and for the new mesh (5 cells in figure 3).

Since $\mathbf{u}$ is constant, then $\sum_{k\in N^+(j)} l_{j,k}(\mathbf{u}, \mathbf{n}_{j,k}) = -\sum_{k\in N^-(j)} l_{j,k}(\mathbf{u}, \mathbf{n}_{j,k})$. The standard condition (7) for the upwind scheme for the cell T_j thus has the form $\frac{\Delta t}{s_j}\sum_{k\in N^+(j)} l_{j,k}(\mathbf{u}, \mathbf{n}_{j,k}) \leq 1$, that is

$$\frac{\Delta t}{s_j}\left(l_{j,k}(\mathbf{u}, \mathbf{n}_{j,k}) + l_{j,l}(\mathbf{u}, \mathbf{n}_{j,l})\right) \leq 1. \tag{14}$$

The CFL condition for the sub-cells $T_{j,k}$ and $T_{j,l}$ are respectively

$$\frac{\Delta t}{s_{j,k}} l_{j,k}(\mathbf{u}, \mathbf{n}_{j,k}) \leq 1 \quad \text{and} \quad \frac{\Delta t}{s_{j,l}} l_{j,l}(\mathbf{u}, \mathbf{n}_{j,l}) \leq 1. \tag{15}$$

Let $l_j = \text{length}\left(\overline{T_{j,k}} \cap \overline{T_{j,l}}\right)$ be the length of the segment separating $T_{j,k}$ and $T_{j,l}$. One has

$$s_{j,k} = \frac{l_j}{2\,|\mathbf{u}|} l_{j,k}(\mathbf{u}, \mathbf{n}_{j,k}), \quad s_{j,l} = \frac{l_j}{2\,|\mathbf{u}|} l_{j,l}(\mathbf{u}, \mathbf{n}_{j,l}) \tag{16}$$

and $s_j = s_{j,k} + s_{j,l} = \frac{l_j}{2|\mathbf{u}|}\left(l_{j,k}(\mathbf{u}, \mathbf{n}_{j,k}) + l_{j,l}(\mathbf{u}, \mathbf{n}_{j,l})\right)$. The two inequalities of (15) and inequality (14) thus rewrite

$$|\mathbf{u}|\,\frac{2\Delta t}{l_j} \leq 1. \tag{17}$$

They are equivalent. It ends the proof.

3.2 Second step: design principle in 1D

The second part of the algorithm is more traditional. Ideally, we just transport the reconstructed profile. We detail the idea in the case of a cell having one upwind neighbor, see figure 3. In this case one has locally, e.g. in the cut cell T_j of figure 3, only *one-dimensional* problems. More precisely, $(T_m, T_{j,k}, T_k)$ and $(T_m, T_{j,l}, T_l)$ are one-dimensional triplet in the sense that

- $T_{j,k}$ has *one* upwind neighbor: T_m, and *one* downwind neighbor: T_k,
- $T_{j,l}$ has *one* upwind neighbor: T_m, and *one* downwind neighbor: T_l.

To perform the numerical advection of the (transversally) reconstructed data, one thus can think of any 3-point transport scheme. The numerical transport can be done with a first order accurate transport scheme (e.g. the upwind scheme; note that the resulting scheme would not be the upwind scheme because of the transverse reconstruction). One can also use any other algorithm: we choose a first-order accurate anti-dissipative method, namely the limited downwind scheme (or Ultra-Bee limiter), which can be reinterpreted as a discontinuous reconstruction scheme, see [20]. Essentially one can use any linear or non-linear scheme in 1D, provided it respects some maximum principle.

Let us switch to 1D notations, table 1. We replace $(c_m, c_{j,k}, c_k)$ or $(c_m, c_{j,l}, c_l)$ by the triplet (c_{i-1}, c_i, c_{i+1}). The one-dimensional velocity is $|\mathbf{u}|$. The global CFL condition is (17), see the proof of lemma 1. It means that the equivalent one-dimensional length associated with sub-cells $T_{j,k}$ and $T_{j,l}$ is $l_j/2$. So the two separate 1D problems will have the same CFL condition.

	c_{i-1}	c_i	c_{i+1}
First 1D problem for cells $(T_m, T_{j,k}, T_k)$	c_m	$c^R_{j,k}$	c_k
Second 1D problem for cells $(T_m, T_{j,l}, T_l)$	c_m	$c^R_{j,l}$	c_l

Table 1: Example of two 1D problems obtained after the reconstruction process applied to the situation depicted in figure 3. The central value c_i is the reconstructed value. The left value c_{i-1} and the right value c_{i+1} are taken from the neighboring cells.

In order to perform the update of the value c_j^n (to compute c_j^{n+1}), one needs to compute the update of $c^R_{j,k}$ and $c^R_{j,l}$, both of them being denoted as c_i^n in the following. It means we have to solve twice the same one-dimensional transport equation $\partial_t c + |\mathbf{u}|\,\partial_x c = 0$. The associated discrete scheme is

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} + |\mathbf{u}|\,\frac{c_{i+1/2}^n - c_{i-1/2}^n}{\Delta x_i} = 0 \text{ with } \Delta x_i = \frac{l_j}{2}.$$

Here $c_{i+1/2}^n$ are the fluxes. Noting $\nu_i = |\mathbf{u}|\,\Delta t/\Delta x_i$, standard fluxes with TVD limitation are $c_{i+1/2}^n = c_i^n + \frac{1}{2}(1-\nu_i)\varphi_{i+1/2}^n\left(c_{i+1}^n - c_i^n\right)$. The $\varphi_{i+1/2}^n$ coefficient is the limiter coefficient. We use the Ultra-Bee limiter

$$\mu_{i+1/2}^n = \frac{1}{2}(1-\nu_i)\varphi_{i+1/2}^n = \text{ Minmod}\left(\frac{(1-\nu_i)(c_i^n - c_{i-1}^n)}{\nu_i(c_{i+1}^n - c_i^n)}, 1\right) \in [0,1]. \qquad (18)$$

The Ultra-Bee limiter it equivalent to a limited downwind scheme, see [7]. This limiter can be interpreted as a particular application of the remark 1 about anti-dissipative schemes see for example [7, 37]. Therefore this limiter is convenient for computations where one seeks strongly optimal non-linear anti-dissipation.

In dimension greater than one, the final flux will be defined as a modification of the reconstructed flux. So the final scheme, written in equation (31), is conservative if $\mathrm{div}\,(\mathbf{u}) = 0$. This fact will be established rigorously in proposition 4.

4 The Vofire scheme

The algebraic extension of the previous algorithm forgets about the geometry of the problem. The mesh is not assumed to be triangular anymore. Let us assume in this section that $\mathbf{u}$ is divergence-free, that is (8) holds true.

4.1 First step

The starting point is the general scheme (5) with the simplification (8). We write

$$s_j \frac{c_j^{n+1} - c_j^n}{\Delta t} + \sum_{k \in N(j)} l_{j,k}(\mathbf{u}_{j,k}, \mathbf{n}_{j,k}) c_{j,k}^R = 0. \tag{19}$$

This is not the final algorithm. The final flux $c_{j,k}^n$ will be defined as the flux $c_{j,k}^R$ (to be defined) plus a correction that will be designed later. For the moment we concentrate on (19). We impose the natural condition $c_{j,k}^R = c_{k,j}^R$. It ensures the conservativity of the scheme when $\mathrm{div}(\mathbf{u}) = 0$.

Remark 2 It is important to notice that $c_{j,k}^R$ is no more related to a reconstruction procedure inside the cell. Instead $c_{j,k}^R$ will be related to the outbound faces. However we state as a design principle that, if the local geometry is the one of figure 3, then the algebraic algorithm shall be equivalent to the geometrical reconstruction procedure that was described in the previous section.

We now describe the method to design these $c_{j,k}^R$. The relations (16) that were derived in section 3.1 show that the conservativity equation (9) can also be rewritten as

$$\sum_{k \in N^+(j)} l_{j,k}(\mathbf{u}_{j,k}, \mathbf{n}_{j,k})(c_{j,k}^R - c_j^n) = 0, \quad \forall j \tag{20}$$

if the velocity is constant $\mathbf{u}_{j,k} = \mathbf{u}$. This constraint (20) is defined on the edges with outbound flux $k \in N^+(j)$. We generalize (20) as a necessary constraint in the general case for a non constant velocity. It means that the outgoing flux before and after reconstruction is supposed to be the same (the upwind value).

To simplify the notations it is convenient to define

$$p_{j,k} = \frac{l_{j,k}(\mathbf{u}_{j,k}, \mathbf{n}_{j,k})}{\sum_{i \in N^+(j)} l_{j,i}(\mathbf{u}_{j,i}, \mathbf{n}_{j,i})} \in [0,1], \quad k \in N^+(j)$$

and

$$p_{j,k} = \frac{l_{j,k}(\mathbf{u}_{j,k}, \mathbf{n}_{j,k})}{\sum_{i \in N^-(j)} l_{j,i}(\mathbf{u}_{j,i}, \mathbf{n}_{j,i})} \in [0,1], \quad k \in N^-(j).$$

One has $\sum_{k\in N^-(j)} p_{j,k} = \sum_{k\in N^+(j)} p_{j,k} = 1$. With these notations, the constraint (20) rewrites

$$\sum_{k\in N^+(j)} p_{j,k} c^R_{j,k} = c^n_j. \tag{21}$$

Then we add a second ingredient, already described in remark 1. It consists in taking $c^R_{j,k}$ as close as possible to the downwind value c^n_k, in order to add anti-dissipation in the scheme. There exist many ways to do that. We propose to minimize the function J

$$J = \sum_{k\in N^+(j)} p_{j,k} \left| c^R_{j,k} - c^n_k \right| \tag{22}$$

with respect to the $c^R_{j,k}$. Define for convenience a parameter $\lambda_{j,k} \in [0,1]$ such that $c^R_{j,k} = c^n_j + \lambda_{j,k}\left(c^n_k - c^n_j\right)$. So $c^R_{j,k}$ is in between c^n_j and c^n_k by construction. This is similar to equation (10). Then $J = \sum_{k\in N^+(j)} p_{j,k} |c^n_k - c^n_j|\,|1-\lambda_{j,k}|$ and also

$$J = \sum_{k\in N^+(j)} p_{j,k} |c^n_k - c^n_j| \left(1-\lambda_{j,k}\right). \tag{23}$$

Constraint (21) rewrites

$$\sum_{k\in N^+(j)} p_{j,k} \left(c^n_k - c^n_j\right) \lambda_{j,k} = 0, \text{ with } \quad \lambda_{j,k} \in [0,1] \ \ \forall k. \tag{24}$$

Proposition 2 The scheme (19) with the constraint (24) preserves the maximum principle under standard CFL condition (7) for divergence-free velocities.

Indeed, the constraint (24) is equivalent to (20). Therefore the scheme (19) rewrites

$$s_j \frac{c^{n+1}_j - c^n_j}{\Delta t} - \sum_{k\in N^-(j)} l_{j,k}(\mathbf{u}_{j,k}, \mathbf{n}_{j,k})(c^n_j - c^R_{j,k}) = 0. \tag{25}$$

Since, by construction, $c^R_{j,k} \in \lfloor c^n_j, c^n_k \rceil$, the maximum principle is a consequence of the CFL condition $\Delta t \sum_{k\in N^-(j)} l_{j,k} \left|(\mathbf{u}_{j,k}, \mathbf{n}_{j,k})\right| \le s_j$.

Let us set

$A_j = \sum_k p_{j,k}(c^n_k - c^n_j)$ where the sum is taken over $k \in N^+(j)$ such that $c^n_k - c^n_j \ge 0$

and

$B_j = \sum_k p_{j,k}(c^n_j - c^n_k)$ where the sum is taken over $k \in N^+(j)$ such that $c^n_j - c^n_k \ge 0$.

That is $N^+(j)$ is split into two sets of indices k: those contributing to A_j and those to B_j. If there is no indices k in the sum that defines A_j (resp. B_j), we set by default $A_j = 0$ (resp. $B_j = 0$). In all cases $A_j \ge 0$ and $B_j \ge 0$.

Proposition 3 The minimum of J under constraints defined in (24) is

$$\min J = \left|A_j - B_j\right|.$$

For the simplicity of notations we skip the index j and define $\alpha_k = p_{j,k}\left(c_k^n - c_j^n\right)$. Then minimization problem is rewritten as: Find the minimum of $J = \sum_k |\alpha_k|(1-\lambda_k)$ under constraints $\sum_k \alpha_k\lambda_k = 0$ and $0 \leq \lambda_k \leq 1$ for all k. Define $0 \leq A = \sum_{\alpha_k \geq 0} \alpha_k$ and $0 \leq B = -\sum_{\alpha_k \leq 0} \alpha_k$. Assume for example that $A \geq B$. Then one has

$$\sum_{\alpha_k \geq 0} \alpha_k\lambda_k = \sum_{\alpha_k \leq 0} (-\alpha_k)\lambda_k \leq \sum_{\alpha_k \leq 0} (-\alpha_k).$$

Since $J = \sum_{\alpha_k \geq 0} \alpha_k - \sum_{\alpha_k \geq 0} \alpha_k\lambda_k - \sum_{\alpha_k \leq 0} \alpha_k(1-\lambda_k)$ then

$$J \geq \sum_{\alpha_k \geq 0} \alpha_k - \sum_{\alpha_k \leq 0} (-\alpha_k) - \sum_{\alpha_k \leq 0} \alpha_k(1-\lambda_k)$$

$$= \sum_{\alpha_k \geq 0} \alpha_k - \sum_{\alpha_k \leq 0} (-\alpha_k)\lambda_k \geq \sum_{\alpha_k \geq 0} \alpha_k - \sum_{\alpha_k \leq 0} (-\alpha_k) = A - B.$$

This value $A-B$ is reached, that is it is the minimum of J. To see this we construct a particular solution. Let us assume that $A > B \geq 0$ (if $A = B = 0$ then $\alpha_k = 0$ for all k so $J = 0$). Let us set $\lambda_k^\star = 1$ if $\alpha_k \leq 0$ and $\lambda_k^\star = \frac{B}{A} \in [0,1[$ if $\alpha_k \geq 0$. So Therefore

$$J(\lambda^\star) = \sum_{\alpha_k \geq 0} \alpha_k \left(1 - \frac{B}{A}\right) = A\left(1 - \frac{B}{A}\right) = A - B = |A - B|$$

which means that the infimum is reached. The other case $B \geq A \geq 0$ is treated with the same method.

Definition 1 The algebraic reconstruction that we use is defined by the minimization of J described in the proof of the previous proposition. (24).

In practice, we use the following procedure where we precise what to do if $A_j = B_j = 0$.

1) If $A_j = 0$ or $B_j = 0$, it means that c_j^n is a local extremum with respect to the c_k^n for $k \in N^+(j)$. Then we do not reconstruct, which means

$$\lambda_{j,k} = 0 \text{ and } c_{j,k}^R = c_j^n \text{ for all } k \in N^+(j). \tag{26}$$

2) If $0 < A_j < B_j$, we take

$$\lambda_{j,k} = 1 \text{ for } k \in N^+(j) \text{ s.t. } c_k - c_j > 0 \tag{27}$$

and

$$\lambda_{j,k} = \lambda_j = \frac{A_j}{B_j} \text{ for } k \in N^+(j) \text{ s.t. } c_j - c_k > 0. \tag{28}$$

3) If $B_j \geq A_j > 0$, this case is symmetric to the previous one. It leads to

$$\lambda_{j,k} = 1 \text{ for } k \in N^+(j) \text{ s.t. } c_j - c_k > 0 \tag{29}$$

and

$$\lambda_{j,k} = \lambda_j = \frac{B_j}{A_j} \text{ for } k \in N^+(j) \text{ s.t. } c_k - c_j > 0. \tag{30}$$

REMARK 3 Assume the geometry is the one of figure 3 (triangle mesh, constant velocity). Then the solution (26-30) of the minimization problem defined on the outbound edges is unique and is equal to the solution (11-13) of the reconstruction procedure inside the cell.

To verify this property, it is sufficient to note that the 3 cases of the above algorithm for finding the solution of the minimization problem are equivalent to the 3 cases described in (11-13). Indeed the $p_{j,k}$ are proportional to the areas $s_{j,k}$, this is a consequence of the relations (16). So in the case the geometry is the one of figure 3, one has the relations $s_{j,k}\left(c_k^n - c_j^n\right) s_{j,l}\left(c_l^n - c_j^n\right) = -\mu A_j B_j$ with $\mu > 0$ and $-\dfrac{s_{j,k}\left(c_k^n - c_j^n\right)}{s_{j,l}\left(c_l^n - c_j^n\right)} = \left(\dfrac{A_j}{B_j}\right)^{\pm 1}$. With these relations the equivalence between (26-30) and (11-13) is immediate.

4.2 Second step

The final scheme is also based on (5) with the simplification (8):

$$s_j \frac{c_j^{n+1} - c_j^n}{\Delta t} + \sum_{k \in N(j)} l_{j,k}(\mathbf{u}_{j,k}, \mathbf{n}_{j,k}) c_{j,k}^n = 0, \tag{31}$$

but with a different flux. The flux $c_{j,k}^n$ for $k \in N^+(j)$ shall be defined as a modification of the reconstructed value $c_{j,k}^R$ by the means of some coefficients $\mu_{j,k,r} \in [0,1]$, $r \in N^-(j)$,

$$c_{j,k}^n = c_{j,k}^R + \left(\sum_{r \in N^-(j)} \mu_{j,k,r} p_{j,r} \right) (c_k^n - c_{j,k}^R). \tag{32}$$

This formula needs some comments and justifications. First, by taking $\mu_{j,k,r} = 0$ for all $j, k \in N(j)$, one recovers $c_{j,k}^n = c_{j,k}^R$. The scheme is then equal to (19) and inherits all its properties. Second, we know that this particular choice is not sufficient for achieving anti-dissipativity, because the definition of $c_{j,k}^R$ corresponds to the first step of the geometric algorithm, the second step is still missing: in 1D, the equivalent scheme would be the upwind scheme.

In the following we mimic the second step of the geometric algorithm and propose a constructive way of achieving stability and anti-dissipativity. In 1D the method leads to a definition of $\mu_{j,k,r} \in [0,1]$ which is equal to $\mu_{i+\frac{1}{2}}$ defined in (18). The reader will also notice that the final formula (38) defines the largest possible $\mu_{j,k,r}$ provided the maximum principle is satisfied. So it is a way to incorporate anti-dissipation in the numerical method, as it was stated in remark 1. By stability, we mean that the scheme must satisfy the maximum principle

$$\min\left(c_j^n, \min_{r \in N^-(j)}(c_r^n)\right) = m_j^n \leq c_j^{n+1} \leq M_j^n = \max\left(c_j^n, \max_{r \in N^-(j)}(c_r^n)\right)$$

under a Courant-Friedrichs-Lewy condition.

The satisfaction of these inequalities is the design principle for the definition of the $\mu_{j,k,r}$. The scheme rewrites

$$c_j^{n+1} = c_j^n - \nu_j \sum_{k\in N^+(j)} p_{j,k} c_{j,k}^n + \nu_j \sum_{r\in N^-(j)} p_{j,r} c_{j,r}^n, \tag{33}$$

where ν_j is the local Courant number (7). By definition of the flux (32), one has the identity

$$\begin{aligned}\sum_{k\in N^+(j)} p_{j,k} c_{j,k}^n &= \sum_{k\in N^+(j)} p_{j,k} \left(c_{j,k}^R + \left(\sum_{r\in N^-(j)} \mu_{j,k,r} p_{j,r} \right) (c_k^n - c_{j,k}^R) \right) \\ &= c_j^n + \sum_{k\in N^+(j)} p_{j,k} \left(\sum_{r\in N^-(j)} \mu_{j,k,r} p_{j,r} \right) (c_k^n - c_{j,k}^R)\end{aligned}$$

since $\sum_{k\in N^+(j)} p_{j,k} c_{j,k}^R = c_j^n$. Plugging this identity in (33) and using $\sum_{k\in N^-(j)} p_{j,k} = \sum_{k\in N^+(j)} p_{j,k} = 1$, one obtains

$$c_j^{n+1} = \sum_{k\in N^+(j),\, r\in N^-(j)} p_{j,k} p_{j,r} \left((1-\nu_j) c_j^n + \nu_j c_{j,r}^n - \nu_j \mu_{j,k,r} (c_k^n - c_{j,k}^R) \right). \tag{34}$$

We observe that c_j^{n+1} is a convex combination with coefficients $p_{j,k}p_{j,r}$ since

$$\sum_{k\in N^+(j),\, r\in N^-(j)} p_{j,k} p_{j,r} = 1.$$

Let us introduce

$$m_{j,r} = \min(c_j^n, c_{j,r}^R) \geq m_j^n \text{ and } M_{j,r} = \max(c_j^n, c_{j,r}^R) \leq M_j^n. \tag{35}$$

We will define the coefficients $\mu_{j,k,r}$ in such a manner that they satisfy the condition

$$m_{j,r} \leq (1-\nu_j) c_j^n + \nu_j c_{j,r}^n - \nu_j \mu_{j,k,r} (c_k^n - c_{j,k}^R) \leq M_{j,r}. \tag{36}$$

In this case the maximum principle is a direct consequence of (34).

Let us provide further details for the definition of $\mu_{j,k,r}$. As $m_{j,r} \leq c_{j,r}^n \leq M_{j,r}$, the situation is quasi one-dimensional in the sense that standard one-dimensional limiters do satisfy these last inequalities. The preceding inequalities (36) are satisfied as soon as $m_{j,r} \leq (1-\nu_j) c_j^n + \nu_j m_{j,r} - \nu_j p_{j,r} \mu_{j,k,r} (c_k^n - c_{j,k}^R)$ and $(1-\nu_j) c_j^n + \nu_j M_{j,r} - \nu_j p_{j,r} \mu_{j,k,r} (c_k^n - c_{j,k}^R) \leq M_{j,r}$ since by construction $m_{j,r} \leq c_{j,r}^n \leq M_{j,r}$. Then: either $c_k^n - c_{j,k}^R > 0$, and the two inequalities are equivalent to $\mu_{j,k,r} \leq \frac{1-\nu_j}{\nu_j} \times \frac{c_j^n - m_{j,r}}{c_k^n - c_{j,k}^R}$; or, on the contrary, $c_k^n - c_{j,k}^R < 0$, and $\mu_{j,k,r} \leq \frac{(1-\nu_j)(c_j^n - M_{j,r})}{\nu_j (c_k^n - c_{j,k}^R)}$. The following inequality gathers these two cases

$$\mu_{j,k,r} \leq \frac{1-\nu_j}{\nu_j} \max \left(\frac{c_j^n - m_{j,r}}{c_k^n - c_{j,k}^R}, \frac{c_j^n - M_{j,r}}{c_k^n - c_{j,k}^R} \right). \tag{37}$$

Following remark 1, maximizing the anti-dissipativity of the resulting scheme is achieved by choosing the largest $\mu_{j,k,r} \in [0,1]$ that satisfies inequality (37). This leads to the final formula

$$\mu_{j,k,r} = \min\left(\frac{1-\nu_j}{\nu_j}\max\left(\frac{c_j^n - m_{j,r}}{c_k^n - c_{j,k}^R}, \frac{c_j^n - M_{j,r}}{c_k^n - c_{j,k}^R}\right), 1\right). \tag{38}$$

Note the similarity with the classical one-dimensional limiter formalism [21, 33, 7]. This choice is used for the test-cases presented in the numerical experiments.

Proposition 4 The VOFIRE scheme (31) is conservative for all values of the fluxes $c_{j,k}^n$. (Evident).

4.3 Interpretation of (36)

It is possible to rewrite inequality (36) in a more convenient and direct way. Indeed let us define $d_{j,k,r}$, for $k \in N^+(j)$ and $r \in N^-(j)$, as the solution of

$$s_j \frac{d_{j,k,r} - c_j^n}{\Delta t} + \left(\sum_{q\in N^+(j)} l_{j,q}(\mathbf{u}_{j,q}, \mathbf{n}_{j,q})\right) d_{j,k,r}^* + \left(\sum_{q\in N^-(j)} l_{j,q}(\mathbf{u}_{j,q}, \mathbf{n}_{j,q})\right) c_{j,r}^n = 0 \tag{39}$$

where the unknown flux is $d_{j,k,r}^* = c_j^n + \mu_{j,k,r}(c_k^n - c_{j,k}^R)$. The quantity $d_{j,k,r}$ is simply the solution of the scheme with the assumption that all fluxes are equal to the same value $c_{j,r}^n$ for $k \in N^-(j)$ and that all fluxes are equal to the same value $d_{j,k,r}^*$ for $k \in N^+(j)$.

Let us rewrite (39)

$$d_{j,k,r} = c_j^n - \nu_j d_{j,k,r}^* + \nu_j c_{j,r}^n = (1-\nu_j)c_j^n + \nu_j c_{j,r}^n - \nu_j \mu_{j,k,r}(c_k^n - c_{j,k}^R).$$

So (36) is equivalent to the simplified problems $m_{j,r} \le d_{j,k,r} \le M_{j,r}$. The number of such simplified problems is $\mathrm{card}(N^+(j)) \times \mathrm{card}(N^-(j))$. Once the maximal value of $\mu_{j,k,r}$ which satisfies the local maximal principle $m_{j,r} \le d_{j,k,r} \le M_{j,r}$ has been computed, one simply combines these fluxes $\mu_{j,k,r}(c_k^n - c_{j,k}^R)$ with the $p_{j,r}$ coefficients. It gives (32). This method is easy to implement.

5 Numerical results for pure transport

We first run a basic test to assess the correctness of each phase of Vofire. We compare the results of the upwind method, the Vofire method, the Vofire method without transverse reconstruction but with longitudinal limiter and the Vofire method with transverse reconstruction but without the upwind longitudinal scheme.

Then we compare systematically four schemes on classical test problems. The schemes are the Upwind scheme, a second order Muscl scheme [1], the Vofire scheme and the VOF method. The VOF method is inspired from [36]. The implementation of this method requires a special treatment in 3D for hexahedrons, because it needs to reconstruct an information about an interface in a cell which has warped faces

in the general case (our objective is to evaluate this method in an ALE context). Therefore the VOF method we have used decomposes each hexahedrons into 24 tetrahedrons for which the application of the VOF methodology is much easier. However this techniques is highly non linear and needs a Newton algorithm to be implemented. Some degeneracy is possible. Moreover we use VOF in the context of ALE, that is the implementation must be compatible with the displacement of the mesh in a second stage of the algorithm, after the reconstruction stage. The results is that the version of VOF that we use respects the maximum principle, but up to some small errors. These small errors are visible on the figures. It illustrates the difficulty of implementing VOF on general meshes for ALE computations. On the other hand VOF gives very good results in terms of accuracy of the interface.

We will consider the function to measure the amount of numerical diffusion

$$t \mapsto F(t) = \int c(t,x)(1-c(t,x))dx. \tag{40}$$

For an analytical solution such that $c = 0$ or $c = 1$, then $F \equiv 0$. For a numerical scheme that satisfies the maximum principle, then $F(t) \geq 0$. Therefore this function F serves as a measure of the artificial diffusion of the numerical solution. We will use this procedure to evaluate the numerical diffusion for all test problems.

Then we compare the CPU costs of the Upwind scheme, a second order Muscl scheme, the Vofire scheme and the VOF method.

5.1 A basic test

We consider a solid body rotation problem, $\mathbf{u}(x,y) = (-2\pi y, 2\pi x)$. The initial data is the characteristic function of a disk of radius 0.2 and center $(0.5, 0.7)$.

We plot in figure 4 the results computed with the full upwind scheme, the Vofire scheme, and also the Vofire scheme with $\lambda \equiv 0$ (but $\mu \neq 0$ a priori) and the Vofire scheme with $\mu \equiv 0$ (but $\lambda \neq 0$ a priori). The best result is the one computed with the Vofire algorithm.

We compare the four different evaluation of $F(t)$ in figure 5. One is computed with the upwind scheme and shows the important diffusion of this method, the three others are computed with the Vofire method, with the Vofire method in which the transverse reconstruction is set to zero (Mod2), and with the Vofire method in which the longitudinal scheme (after the transverse reconstruction) is upwind (Mod1).

5.2 Advection of a square (test 1)

We consider a $100 \times 100 \times 2$ regular hexahedral 3D mesh on a domain $[-1,9] \times [-1,9] \times [0,1]$. The initial data is the characteristic function of a square $[0.8, 3.8] \times [0.8, 3.8] \times [0,1]$. The advection direction $\mathbf{u} = (0,1,0)$ is aligned with one direction of the mesh. Therefore the problem is 1D but computed in 3D configuration. The Courant number is 0.08. The final time is $T = 8$. At $t = 4$ we inverse the velocity field, so that the exact solution at T is equal to the initial solution at $t = 0$. We use this method for all test cases.

The results are displayed in figure 6. The Upwind scheme is very dissipative as usual. The result obtained with a second order MUSCL type method is better, but

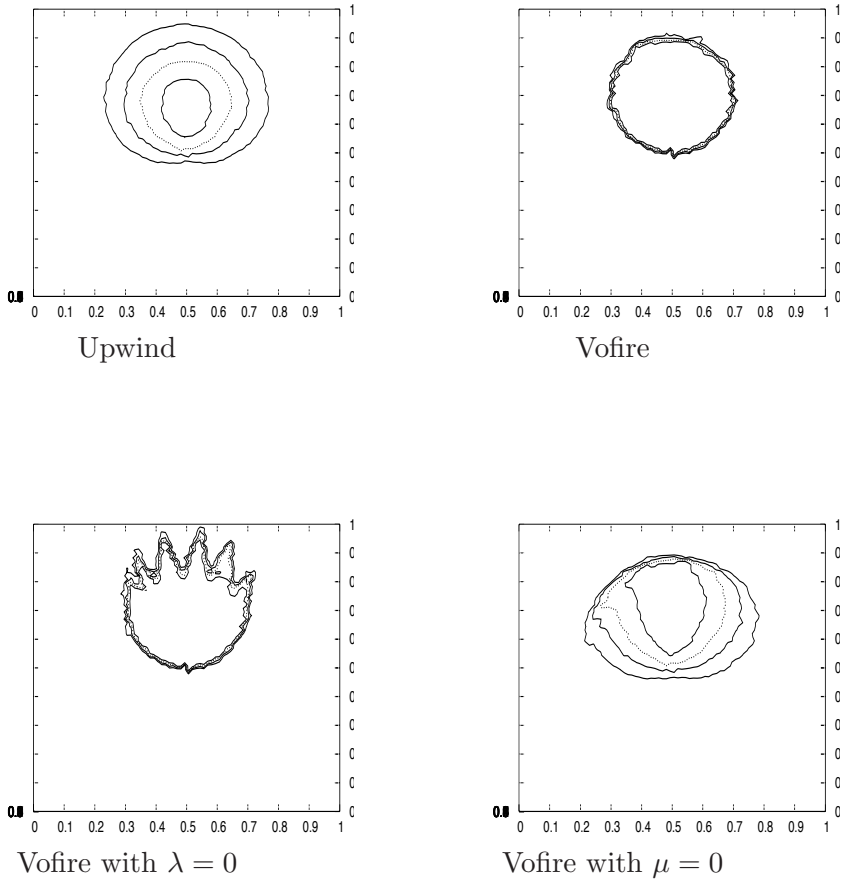


Figure 4: The Vofire method is able to capture the disk after one turn. Other methods incorporate some degree of upwinding. Vofire with $\lambda \equiv 0$ means no transverse reconstruction, but limiter in the longitudinal direction. Vofire with $\mu \equiv 0$ means Upwind after the transverse reconstruction. Computation done on a triangular unstructured mesh (5996 cells), the Courant number is 0.1 (the Courant number is the maximum over all the cells of the local Courant number defined in (7)). The iso-lines are for $c = 0.2, 0.4, 0.6$ and $c = 0.8$.

still with plenty of numerical diffusion. For this problem Vofire is exact, see a proof in [7, 37]. The VOF method is exact in theory. But implementation details make impossible to obtain the full accuracy for reasons that have been explained at the beginning of this section. A common cut shows that Vofire and VOF give very close results. We also plot the value of $F(t)$ where F is computed with the formula (40). It shows that the dissipation of VOF and Vofire is uniformly bounded.

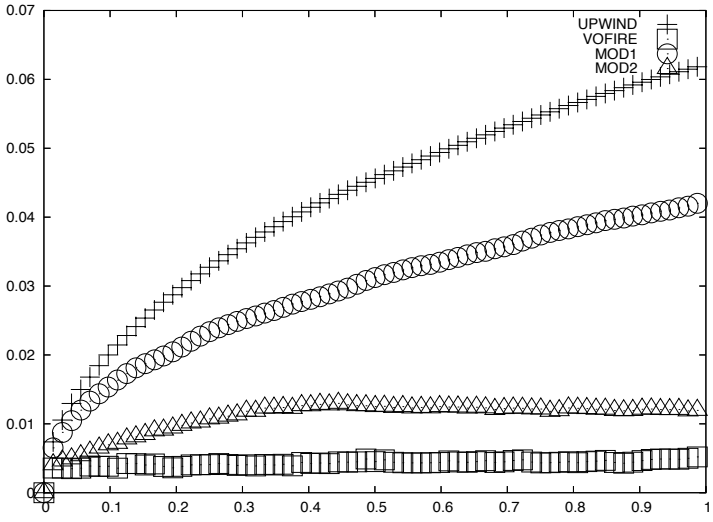


Figure 5: The function $t \mapsto F(t)$ for different schemes. The best curve is the one given by Vofire. Computation done on a triangle unstructured mesh (5996 cells), the Courant number is 0.1. Mod1 (resp. Mod2) corresponds bottom left (resp. bottom right) of figure 4.

5.3 Advection of a square in a non trivial direction (test 2)

The mesh is the same as in test 1. The initial data is the characteristic function of the same square. The advection direction $\mathbf{u} = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0)$ is not aligned with the directions of the mesh. Therefore the problem is 2D but computed in a 3D configuration. The Courant number is 0.07. The final time is $T = 10$.

The results are qualitatively comparable to test case 1, except that Vofire gives a staircase-like numerical solution (this is typical of Vofire, advection in diagonal often furnishes this kind of profiles). All methods have their own default: Upwind has too much dissipation, the second order scheme is better but still with too much dissipation, Vofire has this staircase-tendency and VOF does not respect exactly the maximum principle. The cut shows that VOF and Vofire are quite close. The function F is uniformly bounded for VOF and Vofire (VOF is slightly better however).

5.4 Advection of a square in a tetrahedral mesh (test 3)

We consider a tetrahedral 3D mesh (97853 tets) on a domain $[-1, 9] \times [-1, 9] \times [0, 3]$. For this problem, the VOF method does not need the splitting of hexahedrons. The initial data is the characteristic function of a square. The advection direction is $\mathbf{u} = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0)$. Therefore the problem is 3D. The Courant number is 0.09. The final time is $T = 10$.

The results are displayed in figure 8. No method is exact. However the visual quality of Vofire and VOF is very similar. The staircase approximation of the boundaries is no more visible with Vofire. At the same time the problem that VOF

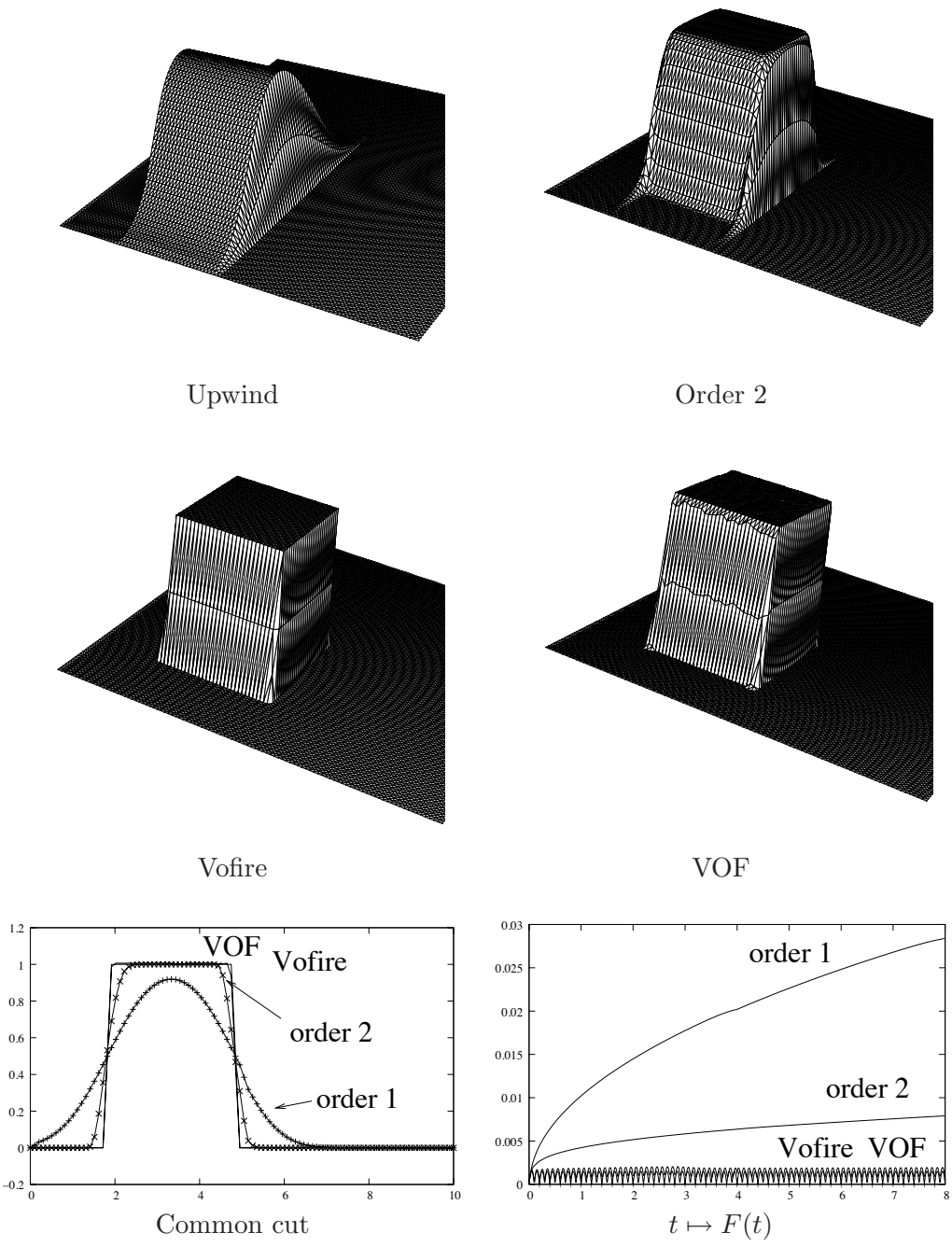


Figure 6: Test 1. Vofire is equal to the exact solution. VOF is almost perfect but the maximum principle is not satisfied because there are small spurious oscillations.

has with the exact preservation of the maximum principle on a hexahedral mesh disappears. The cut and the measure of the dissipation error $t \mapsto F(t)$ show that

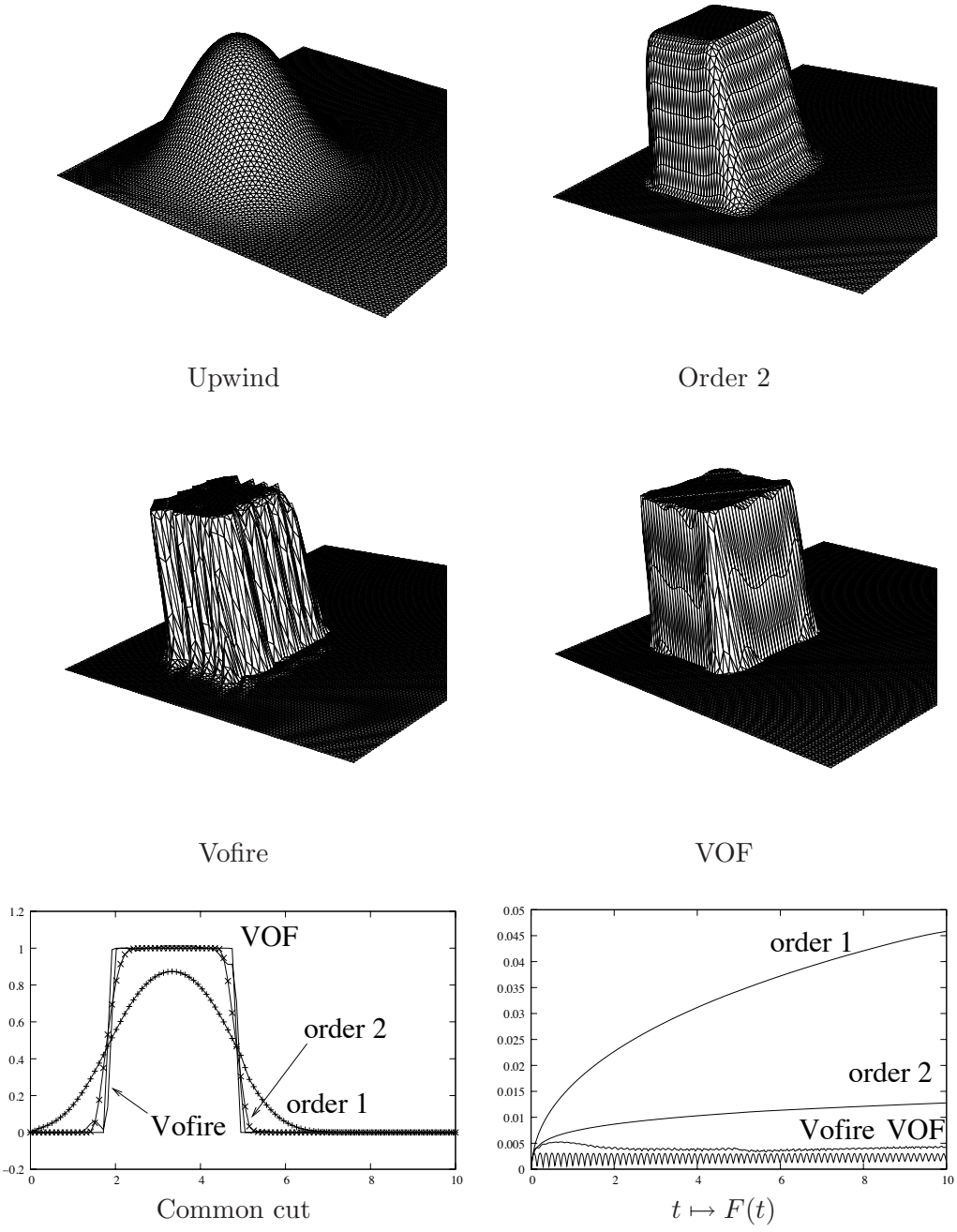


Figure 7: Test 2. Vofire is close to the exact solution but with spurious oscillations. VOF is almost perfect up but the maximum principle is still not satisfied

Vofire and VOF give satisfactory very results.

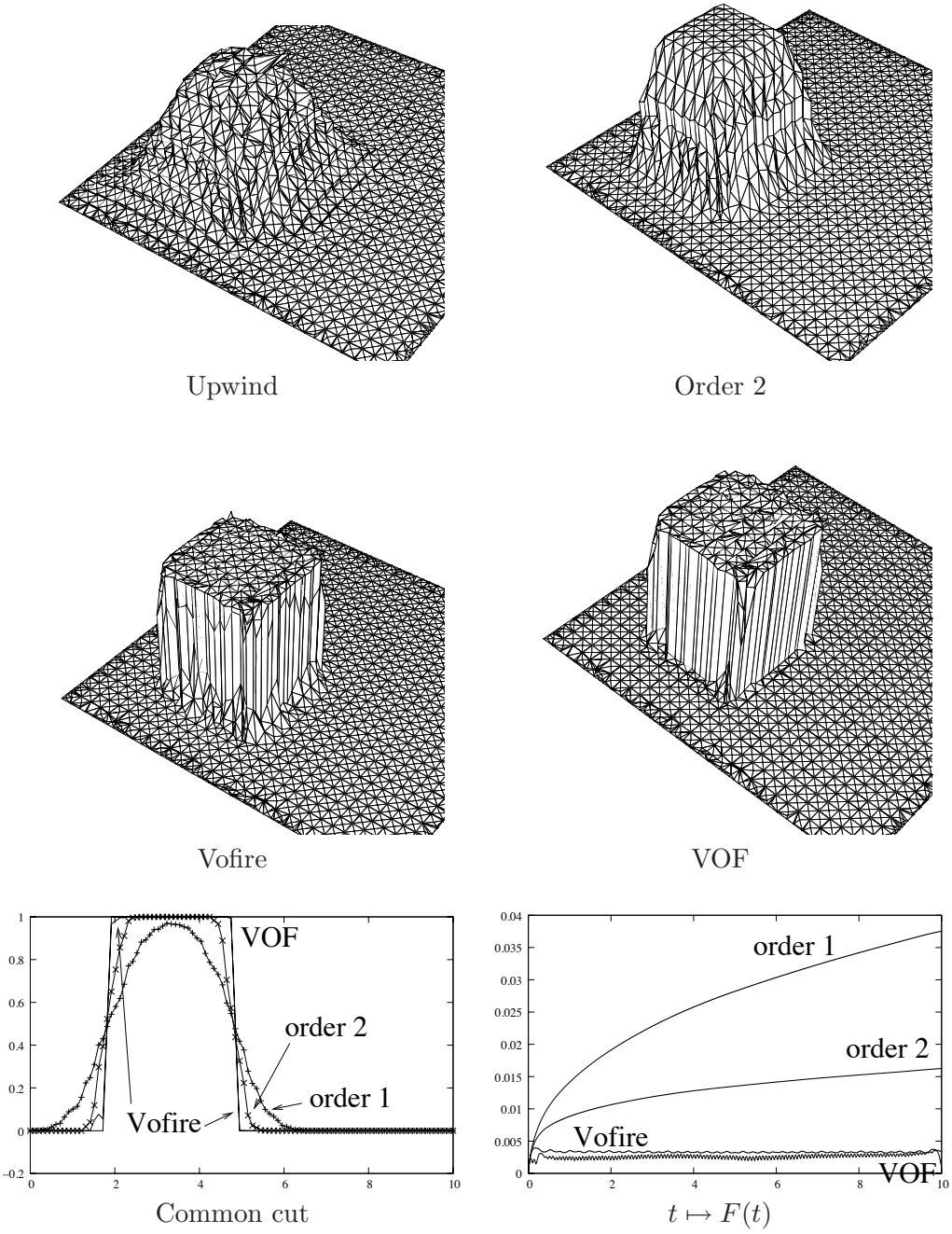


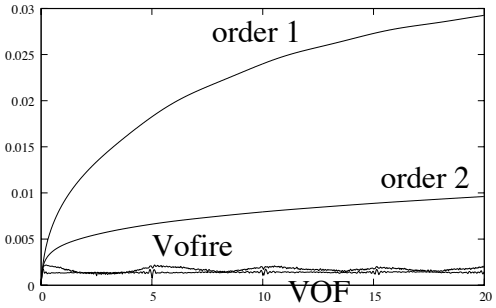
Figure 8: Test 3

5.5 A square in rotation (test 4)

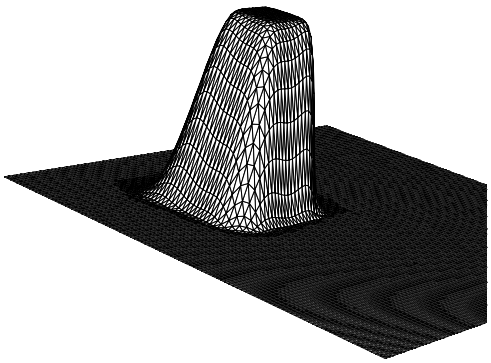
We consider a hexahedral 3D $100 \times 100 \times 2$ mesh of the domain $[-1, 9] \times [-1, 9] \times [0, 1]$. The initial data is the characteristic function of a square. The velocity $\mathbf{u} =$

$\left(\frac{\pi}{10}(x-4), \frac{\pi}{10}(4-y), 0\right)$ is a solid body rotation. The Courant number is 0.15. The final time is $T = 20$.

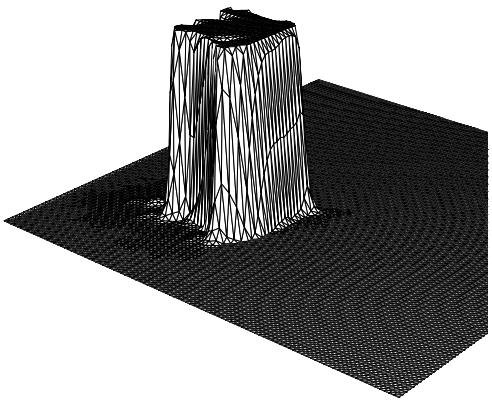
We do not show the cut neither the result with the Upwind scheme, the results are very similar to test cases 1 and 2.



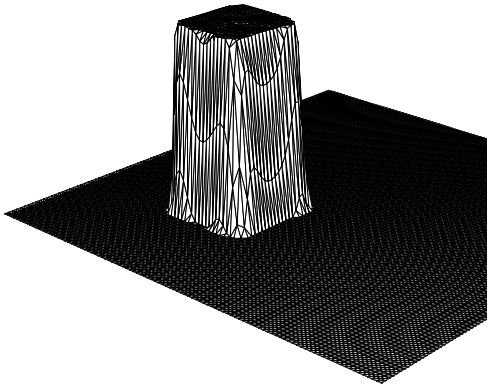
$t \mapsto F(t)$



Order 2



Vofire



VOF

Figure 9: Test 4

5.6 A circle in a vortex flow (test 5)

This problem comes from Rider-Kothe [25]. We consider a hexahedral 3D $100 \times 100 \times 1$ mesh (the mesh is 2D in some sense) of the domain $[-2, 10] \times [-2, 10] \times [0, 1]$.

The initial data is the characteristic function of a circle

$$(x-4)^2+(y-6.5)^2 \leq 1.5^2.$$

The velocity

$$\mathbf{u} = \Big(\cos\frac{\pi}{10}(x-4)\sin\frac{\pi}{10}(y-4), -\sin\frac{\pi}{10}(x-4)\cos\frac{\pi}{10}(y-4), 0\Big)$$

is a vortex flow. The Courant number is 0.08. The final time is $T = 60$.

VOF gives the best result. The quality of the interface is better than with Vofire which is much better than the others results. Upwind is disqualified.

The map $t \mapsto F(t)$ shows that the diffusion reaches a maximum at $t = 30$ for VOF and Vofire. For further times, the diffusion is bounded.

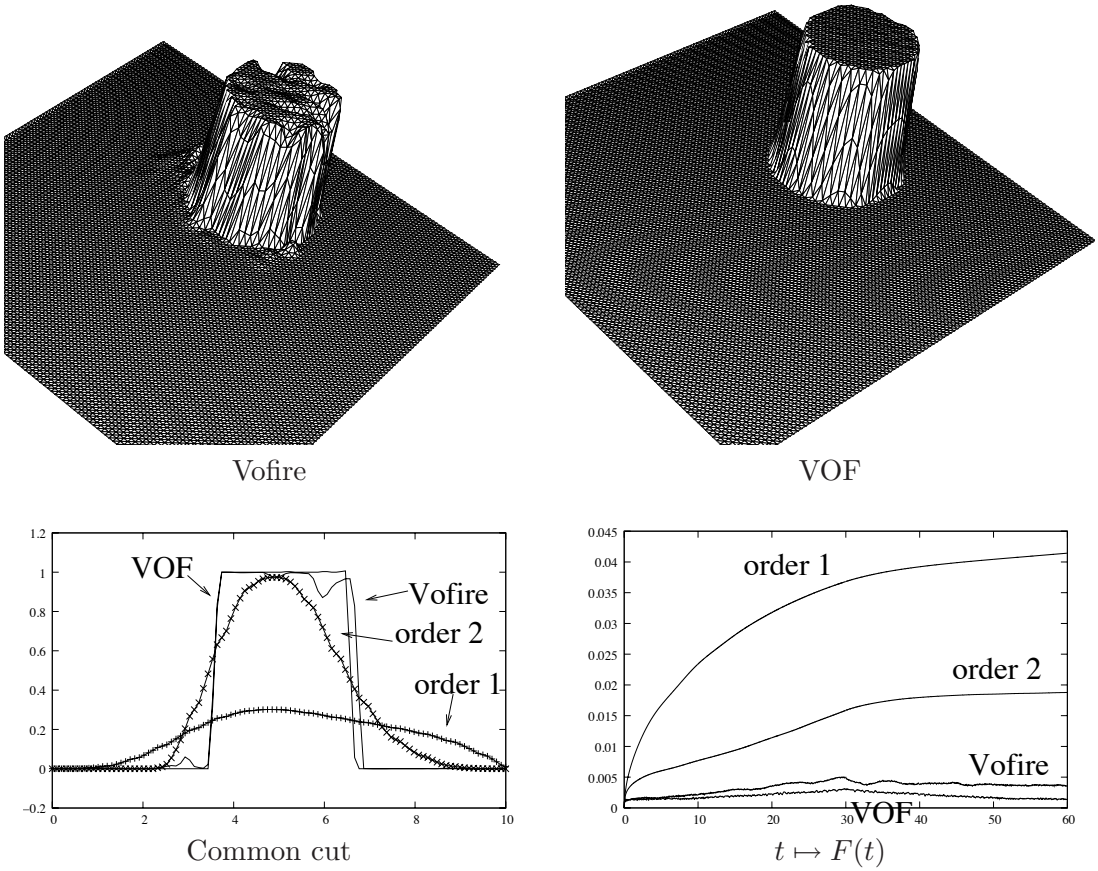


Figure 10: Test 5

5.7 A square in 3D (test 6)

We consider a hexahedral 3D $40\times40\times40$ mesh of the domain $[-1,9]\times[-1,9]\times[-1,9]$. The initial data is the characteristic function of a square. The velocity is $\mathbf{u} =$

$(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$. The Courant number is 0.11. The final time is $T = 10$.

We do not present the results obtained with Upwind and the second order MUSCL scheme. The quality of the interface is quite good with VOF, with a staircase effect with Vofire, see figure 11. The diffusion is better with VOF, but still under control with Vofire. At the same time the diffusion is an increasing function with Upwind and the second order scheme. The cut shows that Vofire displays slights errors at the boundaries of the square. A zoom on the cut profile shows that VOF does not respect the maximum principle due to a very small negative oscillation.

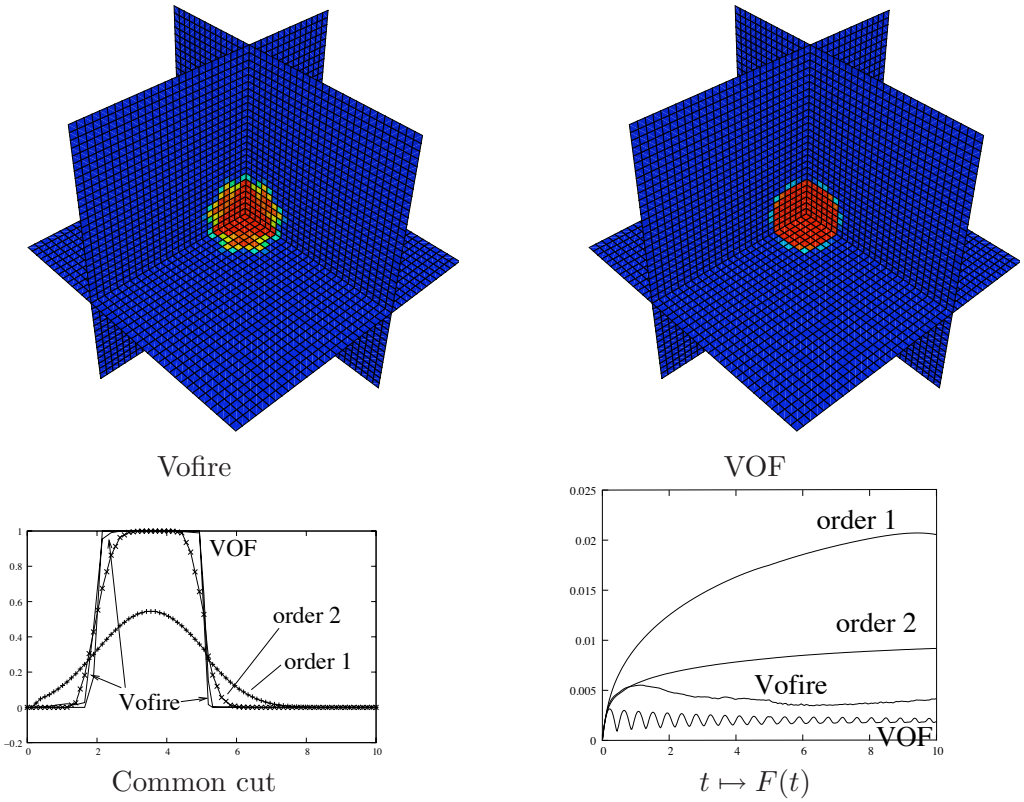


Figure 11: Test 6

5.8 A numerical test of convergence

The Vofire scheme is highly non linear. In consequence theoretical estimates of convergence are out of reach. This is why we propose to examin the following numerical test of convergence, which is a very simple one in dimension two. The domain is the academic square $x = (x_1, x_2) \in]0,1[\times]0,1[$, with periodic boundary conditions. The initial data is

$$c_0(x_1, x_2) = 1 \text{ for } 0 \leq x_1, x_2 \leq 0.5, \text{ and } c_0(x_1, x_2) = 0 \text{ elsewhere.}$$

The velocity is

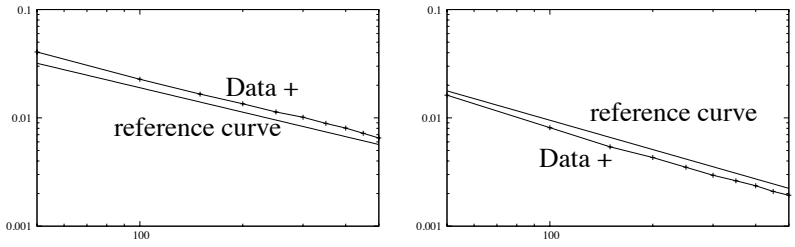
$$\mathbf{u} = (1,1) \text{ for } 0 \leq t < 0.1, \text{ and } \mathbf{u} = -(1,1) \text{ for } 0.1 \leq t < 0.2.$$

cells	Δx	E	F
50×50	0.02	0.04055	0.01620
100×100	0.01	0.02271	0.00811
150×150	0.00666	0.01658	0.00540
200×200	0.005	0.01348	0.00430
250×250	0.004	0.01133	0.00349
300×300	0.00333	0.01013	0.00295
350×350	0.00285	0.00889	0.00262
400×400	0.0025	0.00805	0.00236
450×450	0.00222	0.00720	0.00209
500×500	0.002	0.00652	0.00192
order		≈ 0.75	$\approx .9$

Table 2: The order of convergence of the error E in L^1 is approximatively 0.75. The spreading measure F seems to converge to zero at a higher rate, around .9 for these simulations.

The final time of the simulation if $T = 0.2$ therefore the exact solution at T is equal to c_0. We have performed a series of simulations with finer and finer meshes. At the final time we compute the error in L^1 and we rescale it by the perimeter of c_0. It gives a fist indicator

$$E = \frac{\|c(T) - c_0\|_{L^1(\Omega)}}{|c_0|} = \frac{\|c(T) - c_0\|_{L^1(\Omega)}}{2}.$$



$\Delta x \mapsto E$ $\Delta x \mapsto F$

Figure 12: Convergence slopes: error in L^1 on the left, spreading error F on the right. The reference curve on the left figure is $x \mapsto 0.6x^{-0.75}$. The reference curve on the right figure is $x \mapsto 0.6x^{-0.9}$.

We see that the error in L^1 is of order approximatively 0.75, which can be compared with the order of the upwind scheme which 0.5. By construction this rescaled L^1 error is also an indicator of the number of cells on which the numerical interface is spread. From the values are recorded in table 2, we see that on average

$$2\Delta x \leq E \leq 3\Delta x$$

which means that the numerical interface is spread between two or three cells for this problem. We also compute the spreading indicator

$$F = F(T) = \int_\Omega c(1-c)dx.$$

We observe that $F \approx \Delta x$ which seems to indicate that F converges to zero at a faster rate than E. We have no explanation for this super-convergence behavior.

Finally it must be emphasized that the values of this particular convergence test are easy to interpret. For other problems the numerical order of convergence is much more difficult to identify.

5.9 CPU comparisons

We have incorporated the multimaterial Vofire method in the 3D ALE multi-physics Arcane architecture. It makes possible fair CPU comparisons between different schemes. The table 3 gathers results. However the CPU time is highly dependant on the conditions (IO, number of jobs on the same processor, ...) on which the run has been completed in our institution. So these results must be taken only as a tendency.

The analysis of the results show that the geometrical structure of the VOF method on an hexahedral mesh increases a lot the cost of this method. For test case 3 which uses a tetrahedral mesh, then the CPU cost of VOF is better then Vofire: for this test case the cost is relatively similar compared with the second order Muscl scheme (the Upwind scheme give slightly similar results). It is an indication that the intrinsic CPU cost of VOF and Vofire is under control on tetrahedral meshes, except that the cost needed to cut hexahedrons into 24 tetrahedrons (needed for VOF) makes VOF much more expensive than any other methods in this case.

On the other hand the CPU cost of Vofire is of the same order of magnitude than Upwind and the second order scheme for all meshes. Since the numerical results are better with Vofire (see test cases 1 to 6) then it shows than Vofire is a credible alternative at reasonable cost to enhance the numerical quality for the transport of characteristic functions in various velocity fields.

test case	VOF	Vofire	Order 2
1 (hexs)	6012	1780	823
2 (hexs)	8365	1630	754
3 (tets)	26313	31342	18348
4 (hexs)	27108	2428	1285
5 (hexs)	19297	3282	2308
6 (hexs)	3272	897	446

Table 3: CPU costs in seconds for the test cases

6 Extension to multicomponent Euler equations with non divergent velocity field

The goal is to adapt the previous algorithm to the construction of anti-dissipative fluxes for multimaterial compressible flows such that the velocity field has a priori no reason to be divergence free: that is $\mathrm{div}(\mathbf{u}) \neq 0$.

We focus only on the remapping stage, which follows the Lagrange stage in a Lagrange+remap algorithm, as in [2, 8]. The Lagrange step presents specific difficulties and techniques that we do not discuss. To present the main ideas it is sufficient to focus on the concentration equations of a two-fluid problem,

$$\partial_t(\rho c_1) + \mathrm{div}(\rho c_1 \mathbf{u}) = 0 \text{ and } \partial_t(\rho c_2) + \mathrm{div}(\rho c_2 \mathbf{u}) = 0. \tag{41}$$

Here c_1 and c_2 are the mass fractions, which verify $c_1 + c_2 = 1$. In a Lagrange scheme the mass of each fluid is constant during the time step, therefore the mass fraction of each material is also constant. However a pure Lagrange code is not always possible in 2D and 3D, due to tangling of the mesh. So one usually performs a projection on a new mesh which is close to the previous one, except for the pathological cells. Our objective is the adaptation of the method presented in the previous section to the design of an anti-dissipative multimaterial algorithm in the remap step.

We consider that the remapping is driven by some ALE method, the connectivity of the mesh is preserved but the location of nodes is changed in order to enhance the quality of the mesh. An example is given in figure 13 where a quad is remapped onto a new quad. The shadow zone gives the area that is lost by the cell across the edge. An evaluation of the area of this shadow zone is $\Delta t \times l \times (\mathbf{u}, \mathbf{n})$ where l is the length of the edge, $\mathbf{n}$ is the outward normal and $\mathbf{u}$ is the local edge velocity. The sign of the dot product $(\mathbf{u}, \mathbf{n})$ determines the sign of the swept region area. A priori the edge velocity $\mathbf{u}$ is the half sum of the vertices velocities.

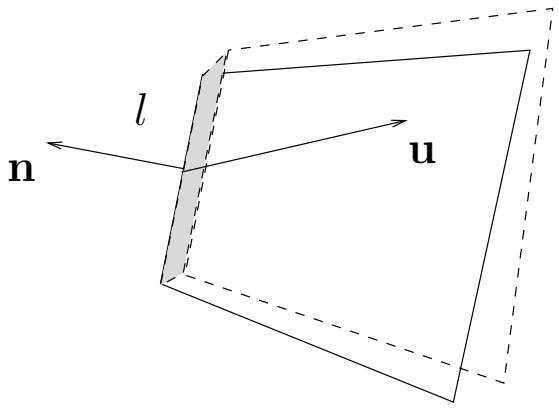


Figure 13: The sweep method and determination of the shadow zone. Area of the shadow zone is $\Delta t \times l \times (\mathbf{u}, \mathbf{n})$. The edge velocity $\mathbf{u}$ is the half sum of the vertices velocities.

In our simulations it is important to control the numerical diffusion of the species. This is why we have chosen to focus on the adaptation of the previous anti-dissipative

algorithm for the mass fractions c_1 and c_2 and not for the total density ρ. We do not assume that $\mathrm{div}(\mathbf{u}) \neq 0$, that is $\sum_k l_{j,k}\,(\mathbf{u}_{j,k}, \mathbf{n}_{j,k}) \neq 0$.

6.1 Vofire for multimaterial computations

Let us present a very simple way to incorporate the Vofire algorithm in the remapping stage based on the sweep method. An important idea is to decouple the projection of the total density, which is done in our case with a standard second order method, from the projection of the concentrations which is done with the Vofire method. The goal is to get the simplest possible method, unchanging the existing hydro-code.

For a two-material problem, the algorithm may be written as

1) First we use a standard second order method for the projection of the total density. That is

- 1-1) We consider that the displacement of the vertices of the mesh is given by a specific algorithm, see for example [12, 32] . So one computes the new area of the cell ${s_j}^{n+1}$. A priori ${s_j}^{n+1} \neq s_j^n$. Then for all interfaces, one can compute the edge velocity with the sweep algorithm. It gives $\mathbf{u}_{j,k}$ for all j, k.
- 1-2) For all cells, determine the inbound and outbound faces in function of the sign of $(\mathbf{u}_{j,k}, \mathbf{n}_{j,k})$. A face incoming into one cell is necessarily outgoing on the other side.
- 1-3) Compute some second order fluxes $\rho_{j,k}^{2\mathrm{nd}}$ for the total density with a standard second order scheme.

At the end of the first part of the method, one knows the total density in each cell:

$$ {s_j}^{n+1}{\rho_j}^{n+1} - s_j^n \rho_j^n + \Delta t \sum_k l_{j,k}\,(\mathbf{u}_{j,k}, \mathbf{n}_{j,k})\,\rho_{j,k}^{2\mathrm{nd}} = 0. \tag{42} $$

We introduce the notations $L_j^{\pm} = \sum_{k \in N^{\pm}(j)} l_{j,k}\,(\mathbf{u}_{j,k}, \mathbf{n}_{j,k})\,\rho_{j,k}^{2\mathrm{nd}}$. So L_j^+ (resp. L_j^-) is the total outgoing (resp. incoming) mass flux.

2) Then construct the concentration fluxes with Vofire as follows

- 2-1) For all cells, perform a transverse reconstruction of the concentration $c = c_1$ using (23) with the constraints and (24). The coefficients $p_{j,k}$ are replaced with the $\widetilde{p_{j,k}}$ $\widetilde{p_{j,k}} = \frac{l_{j,k}\rho_{j,k}^{2\mathrm{nd}}\,(\mathbf{u}_{j,k}, \mathbf{n}_{j,k})}{L_j^+} \in [0,1]$ for $k \in N^+(j)$. The weights $\rho_{j,k}^{2\mathrm{nd}}$ are natural in the sense that the functional J is a measure of the mass flux through the interfaces. This functional to be minimized is

$$ J(\lambda_{j,k}) = \sum_{k \in N^+(j)} \widetilde{p_{j,k}} |c_k^n - c_j^n|\,(1 - \lambda_{j,k}) \tag{43} $$

with the constraint $0 \leq \lambda_{j,k} \leq 1$ for all k and the linear constraint

$$\sum_{k \in N^+(j)} \widetilde{p_{j,k}} \left(c_k^n - c_j^n\right) \lambda_{j,k} = 0. \tag{44}$$

This is solved for all j with the method presented in section 4. At the end of this stage one knows the reconstructed concentration $c_{j,k}^R = c_j^n + \lambda_{j,k}(c_k^n - c_j^n)$ for all $k \in N^+(j)$.

- 2-2) For all cells perform a second limitation of the concentration $c = c_1$ using formulation (39). That is we consider for all $k \in N^+(j)$ and all $r \in N^-(j)$ a prediction $d_{j,k,r}$

$$s_j^{n+1} \rho_j^{n+1} d_{j,k,r} - s_j^n \rho_j^n c_j^n + \Delta t L_j^+ d_{j,k,r}^* + \Delta t L_j^- c_{j,r}^n = 0. \tag{45}$$

This prediction $d_{j,k,r}$ corresponds to a simplified problem. The outgoing flux is $d_{j,k,r}^* = c_j^n + \mu_{j,k,r}\left(c_k^n - c_{j,k}^R\right)$. We compute the coefficient $\mu_{j,k,r} \in [0,1]$ such that the prediction satisfies $d_{j,k,r} \in [m_{j,r}, M_{j,r}]$ where $m_{j,r} = \min\left(c_j^n, c_r^n\right)$ and $M_{j,r} = \max\left(c_j^n, c_r^n\right)$. There is always a trivial solution[1]: $\mu_{j,k,r} = 0$. Non trivial solution will be used to incorporate anti-dissipation. After that we compute the real flux

$$c_{j,k}^n = c_{j,k}^R + \left(\sum_{r \in N^-(j)} \mu_{j,k,r} \widetilde{p_{j,r}}\right) (c_k^n - c_{j,k}^R), \tag{46}$$

- 2-3) Compute $c_2 = 1 - c_1$.

3) Once this has been done, remap the concentrations $c = c_1$ and $c = c_2$ with

$${s_j}^{n+1} {\rho_j}^{n+1} c_j^{n+1} - s_j^n \rho c_j^n + \Delta t \sum_k l_{j,k}\left(\mathbf{u}_{j,k}, \mathbf{n}_{j,k}\right) \rho_{j,k}^{2^{\text{nd}}} c_{j,k}^n = 0. \tag{47}$$

The equations for the definition of the Vofire flux are symmetric for the transformation $c \mapsto 1-c$. Therefore for a two-material problem one has by construction $(c_1)_{j,k}^n + (c_2)_{j,k}^n = (c_1)_j^n + (c_2)_j^n = 1$. So equations (42)-(47) are compatible since $(c_1)_j^{n+1} + (c_2)_j^{n+1} = 1$.

6.2 Stability analysis for 2 materials

Despite the apparent complexity of this method, its stability analysis is easy to conduct following the method used in section 4. In practice it is sufficient to prove that the partial masses are non-negative. For simplicity we first detail the analysis for the two-material case.

[1] Assume that $\mu_{j,k,r} = 0$. Then (42)-(45) imply

$$d_{j,k,r} = \left(\frac{s_j^n \rho_j^n}{{s_j}^{n+1} {\rho_j}^{n+1}} - \frac{\Delta t}{{s_j}^{n+1} {\rho_j}^{n+1}} L_j^+\right) c_j^n - L_j^- c_{j,r}^n.$$

In any cases $c_{j,r}^n \in [m_{j,r}, M_{j,r}]$ by (46) for $r \in N^-(j)$. Therefore $d_{j,k,r}$ is a convex combination of $c_{j,r}^n$ and $c_{j,r}^n$. So $d_{j,k,r} \in [m_{j,r}, M_{j,r}]$ is also true for all $c_{j,r}^n \in [m_{j,r}, M_{j,r}]$. It proves that $\mu_{j,k,r} = 0$ is a trivial solution.

6.2.1 Transverse reconstruction

In order prove the stability, we begin with the case where the flux is $c^R_{j,k}$, ($c = c_1$ or c_2) which means that the longitudinal flux (after reconstruction) is the upwind one: $\mu_{j,k,r} \equiv 0$. Equations (42)-(47) imply

$$c_j^{n+1} = \left(\frac{s_j^n \rho_j^n}{{s_j}^{n+1} {\rho_j}^{n+1}} - \frac{\Delta t}{{s_j}^{n+1} {\rho_j}^{n+1}} L_j^+ \right) c_j^n \tag{48}$$

$$+ \sum_{k \in N^-(j)} \left(\frac{\Delta t}{{s_j}^{n+1} {\rho_j}^{n+1}} l_{j,k} \left(-\mathbf{u}_{j,k}, \mathbf{n}_{j,k} \right) \rho_{j,k}^{2^{\text{nd}}} \right) c^R_{j,k}.$$

Assuming the CFL condition $s_j^n \rho_j^n - \Delta t \sum_{k \in N^+(j)} l_{j,k} \left(\mathbf{u}_{j,k}, \mathbf{n}_{j,k} \right) \rho_{j,k}^{2^{\text{nd}}} \geq 0$, all coefficients are non-negative. Their sum is equal to 1. The non-negativity of the mass fraction c_j^{n+1} for the transverse reconstruction follows. More precisely one has the maximum principle

$$\min \left(c_j^n, \min_{r \in N^-(j)} (c_r^n) \right) = m_j^n \leq {c_j}^{n+1} \leq M_j^n = \max \left(c_j^n, \max_{r \in N^-(j)} (c_r^n) \right). \tag{49}$$

The right and left bounds involve only incoming boundaries.

6.2.2 Complete algorithm

Now we add the second step of the method which means the $\mu_{j,k,r}$ are not necessarily set to zero. By construction each prediction $d_{j,k,r}$ is such that $m_{j,r} \leq d_{j,k,r} \leq M_{j,r}$. Define $d_j = \sum_{k \in N^+(j),\, r \in N^-(j)} p_{j,k} p_{j,r} d_{j,k,r}$. Then one checks from (45) that

$$s_j^{n+1} \rho_j^{n+1} d_j - s_j^n \rho_j^n c_j^n + \Delta t \left(\sum_{q \in N^+(j)} l_{j,q} (\mathbf{u}_{j,q}, \mathbf{n}_{j,q}) \rho_{j,q}^{2^{\text{nd}}} d^*_{j,k,r} \right)$$

$$+ \Delta t \left(\sum_{q \in N^-(j)} l_{j,q} (\mathbf{u}_{j,q}, \mathbf{n}_{j,q}) \rho_{j,q}^{2^{\text{nd}}} c^n_{j,r} \right) = 0.$$

Since $\sum_{q \in N^+(j)} l_{j,q} (\mathbf{u}_{j,q}, \mathbf{n}_{j,q}) \rho_{j,q}^{2^{\text{nd}}} d^*_{j,k,r} = \sum_{q \in N^+(j)} l_{j,q} (\mathbf{u}_{j,q}, \mathbf{n}_{j,q}) \rho_{j,q}^{2^{\text{nd}}} c^n_{j,k}$ then it shows that $c_j^{n+1} = d_j$. Therefore c_j^{n+1} satisfies the local maximum principle.

6.3 Three materials and more

For three materials and more the analysis is a little more tricky. Assume for simplicity that there are only three materials in the simulation. Then one can have

$$(c_1)^n_{j,k} + (c_2)^n_{j,k} + (c_3)^n_{j,k} \neq 1$$

for the Vofire fluxes. It has two implications. First, in equation (48) the sum of the coefficients in parenthesis may be different from 1. Second the equation (42) is not the sum of the three equations (47) for $c = c_1$, $c = c_2$ and $c = c_3$. There is an

incompatibility, which means that we have to reconsider the algorithm. Fortunately the total density computed in (42) is not really important in the definition of the new concentrations (47). It is the product $\rho_j^{n+1}c_j^{n+1}$ that has a physical meaning: it is the new partial density in the cell. So equation (42) is just a prediction of the total density in the cell. Let us analyze the consequences. Now c_j^{n+1} is just a prediction of the new concentration since ρ_j^{n+1} is also a prediction of the new total density in the cell. But what is physically important is the product $\rho_j^{n+1}c_j^{n+1}$. One gets in the cell a numerical value for $\rho_j^{n+1}(c_1)_j^{n+1}$, $\rho_j^{n+1}(c_2)_j^{n+1}$ and $\rho_j^{n+1}(c_3)_j^{n+1}$. Then one defines the partial densities $(\rho_1)_j^{n+1} = \rho_j^{n+1}(c_1)_j^{n+1}$, $(\rho_2)_j^{n+1} = \rho_j^{n+1}(c_2)_j^{n+1}$ and $(\rho_3)_j^{n+1} = \rho_j^{n+1}(c_3)_j^{n+1}$. The new total density is the sum of the partial densities $\left(\rho_j^{n+1}\right)_{true} = (\rho_1)_j^{n+1}+(\rho_2)_j^{n+1}+(\rho_3)_j^{n+1}$. The new concentrations are $\left((c_1)_j^{n+1}\right)_{true} = \frac{(\rho_1)_j^{n+1}}{\left(\rho_j^{n+1}\right)_{true}}\ldots$ Therefore one recovers a global maximum principle $0 \leq \left((c_1)_j^{n+1}\right)_{true} \leq 1\ldots$ Which is sufficient in practice. The local maximum principle (49) could be ensured by enforcing $(c_1)_{j,k}^n + (c_2)_{j,k}^n + (c_3)_{j,k}^n = 1$ with specific methods such as in [14].

7 A numerical result in 3D

We consider a challenging problem, which is representative of the difficulties encountered by other interface reconstruction algorithms: the initial geometry contains a T junction where the three different materials are in contact; the initial density and pressure gradients generate a 3D vortex; it makes impossible the computation of such a flow with a pure Lagrangian scheme and requires specific ALE techniques.

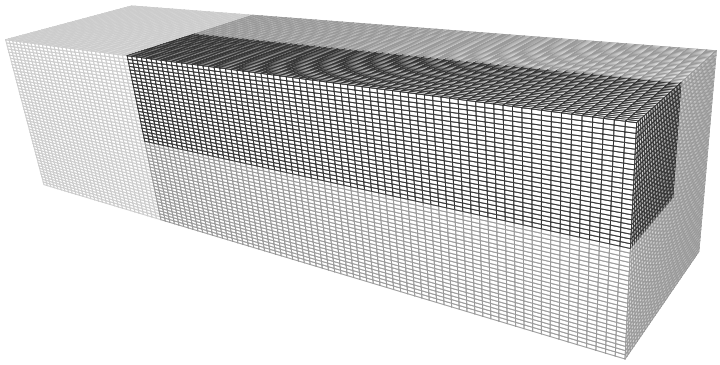


Figure 14: Initial mesh for test case 3.

The dimension of the problem represented in figure 14 in 3D are $[0,1]\times[0.,0.25]\times[0.,0.25]$. In the left part ($x \leq 0.3$), we take $\rho = 1$, $p = 1$ and $c_1 = 1$. In the right interior part ($x \geq 0.3$, $y \leq 0.125$ and $z \leq 0.125$), we take $\rho = 1$, $p = 0.1$ and $c_2 = 1$. In the complementary part the initial data are $\rho = 0.125$, $p = 0.1$and $c_3 = 1$. The initial velocity is $\mathbf{u} = (0,0,0)$ everywhere. The equation of state is a

perfect gas law with $\gamma = 1.4$ (the same for the three parts), so a reference solution of this problem can easily be computed with a standard Eulerian code. For these initial data, the solution at the final time $t = 0.4$ contains a distorted T junction inside the vortex region. A shock is created at the vertical interface, oriented to the right. The results have been computed with ALE techniques, that is the mesh moves. We use the Lagrange scheme described in [9], but the results are quite similar with another Lagrange scheme based on the Von Neumann-Richtmyer scheme [2]. A vortex is created. Therefore the mesh tangles unless specific remeshing algorithms are used. In our case we used a weighted Tipton-Jun method [15, 32] to regularize the mesh, the weights are calculated in order to adapt the mesh around the pressure and concentration gradients.

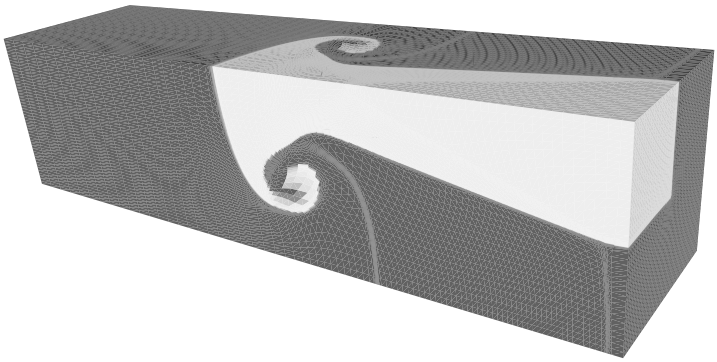


Figure 15: Concentration of the internal block.

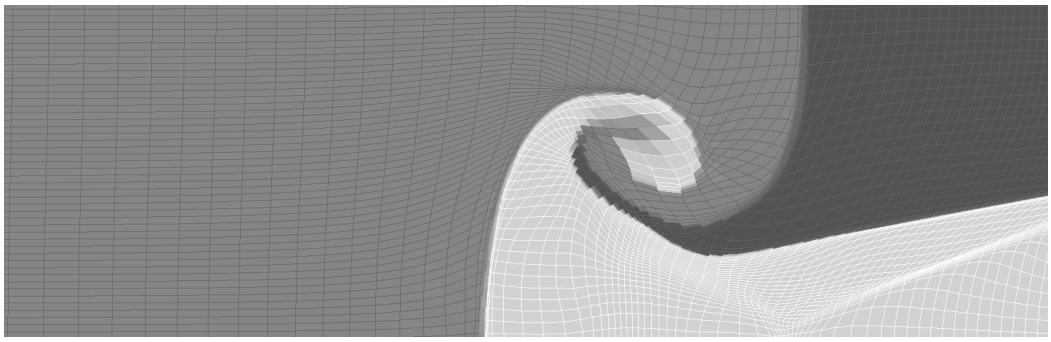


Figure 16: The mesh after displacement. Zoom on the top of the 3D mesh. The three materials are visible in white, grey and black. The antidissipative feature of Vofire is able to separate the materials. However in the center of the vortex low resolution introduce some spreading.

The concentration of the interior zone is represented in figure 15. The mesh at time $t = 0.4$ and $y = 0.25$ corresponding to the superior plane is represented

on figure 16, together with a map of the concentrations. One clearly sees that the initial T-junction is inside the vortex, which makes this computation challenging. For this we use exactly the algorithm described in section 6. One observes that the concentration profile is steep (even if it is not perfect), especially near the vortex region, thanks to the Vofire algorithm. A similar computation but without any reconstruction or anti-dissipative method leads to an excessive amount of mixing. We have also validated the results by means of a comparison with a reference 3D Eulerian code with a fine mesh [5]. More results can be found in [18].

8 Conclusion

We have presented the Vofire algorithm which is an anti-diffusive scheme for transport on arbitrary meshes and in any dimension. By construction Vofire satisfies the maximum principle for all meshes. At the present time it is not possible to prove rigorously the efficiency of such a method. Therefore we rely on numerical experiments (test-cases) to illustrate the gain of accuracy provided by this scheme. The numerical tests for pure transport show that Vofire is indeed anti-diffusive, in the sense that the function $t \mapsto F(t)$ is uniformly bounded and very close the the same function evaluated with the VOF algorithm. For multimaterial computations, it means that Vofire reduces the influence of mixed cells. We have showed that the CPU cost is in bound with Vofire, which is not the case for VOF on an hexahedral mesh (with warped faces).

Even if it was not our initial intention, the tests also show that VOF is sensitive to the geometrical method which is used, first to intersect a general hexahedron (with warped faces) by a plane, second to move the global mesh in an ALE context, and third to project the first mesh onto the final one. In practice it is recommended to postprocess VOF with a clipping technique near extremas to guarantee the maximum principle. These issues raise non trivial implementation difficulties, a stressed in [35] for example. On the other hand VOF corresponds somehow to an optimum with respect to the quality of the interface. Additional research material on geometric reconstruction of 3D interface can be found in [22, 30] and references therein.

However the reader must be aware that the VOFIRE algorithm is no a cure for all the failures of transport algorithms, even if it has less technical difficulties than the VOF method. Some dissipation remains, this is clear at the observation of the numerical tests. Some artificial steepening may appear in the simulation. We observe that it is experimentally on bounds. This steepening is somewhat more pronounced for Cartesian meshes for reasons that we ignore.

References

[1] Barth T. and Ohlberger M. (2004)): *Finite volume methods: foundation and analysis*. Encyclopedia of computational mechanics. Stein, de Borst and Hugues Editors. John Wiley and Sons.

[2] D. J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Computer Methods in Applied Mechanics and Engineering 99 (1992), 235–394.

[3] D. J. Benson, Volume of fluid interface reconstruction methods for multi-material problems, Appl. Mech. Rev. (2002), 55:151–165.

[4] F. Bouchut, An antidiffusive entropy scheme for monotone scalar conservation law, Journal of Scientific Computing 21 (2004), pp. 1–30.

[5] M. Boulet, Richtmyer-Meshkov Instability: 3D Numerical Simulations (TRICLADE code) Proceedings of the IWPCTM 10, 127-153, 2006.

[6] Després B. and Lagoutière F. (2001): *Generalized Harten formalism and longitudinal variation diminishing schemes for linear advection on arbitrary grids*. M2AN Math. Model. Numer. Anal. 35, no. 6, pp. 1159–1183.

[7] Després B. and Lagoutière F. (2002): *Contact discontinuity capturing schemes for linear advection and compressible gas dynamics*. J. Sci. Comput,. 16 (2001), no. 4: pp. 479–524.

[8] Després B., Lagoutière F. (2006): *Numerical resolution of a two-component compressible fluid model with interfaces*, Progress in Computational Fluid Dynamics, vol. 7 (6), 295–310 (2007).

[9] Després B. and Mazeran C. Lagrangian gas dynamics in two dimensions and Lagrangian systems, Arch. Ration. Mech. Anal. 178, No. 3, 327-372 (2005).

[10] J. Donea, A. Huerta J. Ph. Ponthot and . Rodriguez-Ferran, ALE Methods, chapter in Encyclopedia of computational mechanics, Editors E. Stein, R. de Borst and T.J.R. Hughes ed., John Wiley & Sons Ltd., 2004.

[11] V. Dyadechko et M. Shashkov, Moment-of-fluid interface reconstruction, Rapport LA-UR-05-7571, LANL, 2005.

[12] P.L. George, H. Borouchaki, P.J. Frey, P. Laug and E. Saltel, Mesh generation and mesh adaptivity: theory, techniques, chapter in Encyclopedia of computational mechanics, Editors E. Stein, R. de Borst and T.J.R. Hughes ed., John Wiley & Sons Ltd., 2004.

[13] C. Hirt, A. Amsden and J. Cook, An arbitrary lagrangian-eulerian computing method for all flow speeds, J. of Comp. Phys., 14:227–253, 1974.

[14] Jaouen S., Lagoutière F., Numerical transport of an arbitrary number of components. Comput. Methods Appl. Mech. Engrg. 196 (2007), no. 33-34, 3127–3140.

[15] B. I. Jun, A modified equipotential method for grid relaxation, LLNL, 2000.

[16] D. B. Kothe, M. W. Williams, K. L. Lam, D. R. Korzekwa and P. K. Tubesing, A second order accurate, linearity-preserving volume tracking algorithm for free surface flows on 3-D unstructured meshes, Proceedings of the 3rd ASME/JSME Joint Fluids Engineering Conference, July 18-22, San Fransisco, California, USA.

[17] D. Kuzmin, R. Lohner and S. Turek (Eds), Flux-corrected transport: principles, algorithms and applications, Springer, Scient. Comp., 2005.

[18] E. Labourasse, B. Després and F. Lagoutière, The Vofire Finite Volume method applied to multicomponent flows on unstrutured meshes, in Finite Volume for complex applicaions V, R. Eymard and J. M. Hérard Editors, Wiley, Problems and perspectives, 2008.

[19] Lagoutière F. (2007): *Stability of reconstruction schemes for scalar hyperbolic conservation laws*, to appear in Communications in Mathematical Sciences.

[20] Lagoutière F. (2007): *Nondissipative reconstruction schemes satisfying entropy inequalities*, preprint.

[21] R.J. LeVeque. (1992): *Numerical methods for conservation laws*. ETHZ Zurich, Birkhauser, Basel.

[22] J. Lopez and J. Hernandez, *Analytical and geometrical tools for 3D volume of fluid methods on general grids*, Journal Comp. Phys., 227, 12, 5939–5948.

[23] Mosso S. and Cleancy S. (1995), *A geometrical derived priority system for Young's interface reconstruction*. LA-CP-95-0081, LANL report.

[24] Noh W. F. and Woodward P. R. (1976), *SLIC (Simple Line Interface Calculation)*. Springer Lecture Notes in Physics, 25: pp. 330–339.

[25] W. J. Rider et D. B. Kothe, Streching and tearing interface tracking methods, AIAA paper, 95, 1-11, 1995.

[26] M. Rudman, Volume tracking methods for interfacial flow calculations, Int. J. Numer. Meth. Fluids, 24, 671-691, 1997.

[27] Scardovelli R, Zaleski S (1999) Direct numerical simulation of free-surface and interfacial flow, Annual Review Of Fluid Mechanics 31: 567-603.

[28] M. Sussman, P. Smereka and S. Osher, A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow Journal of Computational Physics, Volume 114, Issue 1, September 1994, Pages 146-159

[29] H. T. Taek Ahn and M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, Journal of Computational Physics, Volume 226, Issue 2, 1 October 2007, Pages 2096-2132.

[30] H. T. Taek Ahn and M. Shashkov, Geometric algorithms for 3D interface reconstruction, in Proceedings of the 16th International Meshing Roundtable, M. L. Brewer and D. Marcum Editors, 2008.

[31] H. T. Taek Ahn, M. Shashkov and M. A. Christon, The moment-of-fluid method in action, LA-UR-6854, Los Alamos National Laboratory, 2007, to appear in Communications in Numerical Methods for Engineering.

[32] R. Tipton, Grid optimization by equipotential relaxation, Lawrence Livermore National Laboratory, 1992.

[33] E. F. Toro. (1997), *Riemann solvers and numerical methods in fluid dynamics, a practical introduction*, Springer.

[34] S. Unverdi et G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, JCP, 100, 25-37, 1992.

[35] S. Vincent and J.P. Caltagirone, Efficient solving method for unsteady incompressible interfacial flow problems, Int. Jour. for Num. Methods in Fluids, 30, 795–811, 1999.

[36] Youngs D. L., *Numerical methods for fluid dynamics, Time-dependent multimaterial flow with large fluid distortion*. Academic Press, NY, Morton and Baines Editors, 1982.

[37] X. Zhengfu and C. W. Shu, Anti-diffusive flux corrections for high order finite difference WENO schemes, J. Comput. Phys. 205, No. 2, 458-485 (2005). I

[38] Zalesak S. T. (1979), *Fully multidimensional flux-corrected transport algorithms for fluids*, Journal of Computational Physics, 31, pp. 335–362.